VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

DEKÓDOVÁNÍ ČÁROVÉHO KÓDU V OBRAZE V REÁLNÉM ČASE

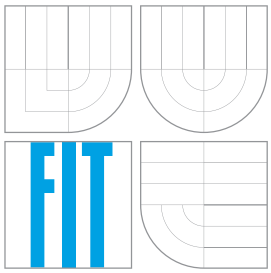
DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. MARTIN KOSTIHA

BRNO 2010

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

DEKÓDOVÁNÍ ČÁROVÉHO KÓDU V OBRAZE V REÁLNÉM ČASE

DECODING BARCODE IN IMAGE IN REAL TIME

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. MARTIN KOSTIHA

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. ADAM HEROUT, Ph.D.

BRNO 2010

Abstrakt

Tato práce se zabývá lokalizací a dekódováním lineárního čárového kódu EAN-13 v obraze. Popisuje typy a vlastnosti čárových kódů a představuje možné přístupy k lokalizaci a dekódování. Praktická část se věnuje návrhu a implementaci algoritmu s ohledem na efektivitu pro zpracování v reálném čase. Lokalizace je založena na hledání podobných rysů čárového kódu mezi čtenými řádky obrazu a dekódvání je prováděno nalezením nejvyšší podobnosti vůči referenčním kódovaným číslicím.

Abstract

This work deals with localization and decoding of the linear barcode EAN-13 in the image. It describes types and qualities of barcodes and pose posibble accesses into the localization and decoding. The practical part concerns with the concept and implementation of the algorithm having regard to effectiveness for processing in real time. The localization is established on finding similar features of the barcode among the read rows of the image and decoding is made by finding of the supreme conformity towards referential code figures.

Klíčová slova

čárový kód, EAN-13, symbolika, lokalizace čárového kódu, dekódování čárového kódu

Keywords

barcode, EAN-13, symbology, barcode localization, barcode decoding

Citace

Martin Kostiha: Dekódování čárového kódu v obraze v reálném čase, diplomová práce, Brno, FIT VUT v Brně, 2010

Dekódování čárového kódu v obraze v reálném čase

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Adama Herouta, Ph.D.

.....
Martin Kostiha
25. května 2010

Poděkování

Děkuji vedoucímu mé diplomové práce Ing. Adamu Heroutovi, Ph.D. za odbornou pomoc a cenné rady.

Obsah

Kapitola 1

Úvod

Čárové kódy jsou strojově čitelnou reprezentací informací ve vizuální podobě. Slouží k automatické identifikaci, jsou velmi populární a staly se všudypřítomné ve všech formách podnikání. Jeden z hlavních důvodů proč jsou čárové kódy dnes široce používány a neustále vyvíjeny je jejich vysoká spolehlivost a možnost zahrnout velké množství informací na velmi malou tiskovou oblast. Náklady na straně označení jsou minimální. Čárový kód může být vytištěn na libovolném levném papíře a může být umístěn téměř na jakýkoliv objekt. S pomocí technologie pro zpracování čárových kódů jsou rychlým a přesným nástrojem pro zadávání vstupních dat bez klávesnice. Pomáhají tedy podnikům minimalizovat chyby při zadávání dat, urychlují proces a snižují náklady po více než 30 let.

Většinou jsou čárové kódy známé z obchodních domů, kde je jimi označováno zboží, ale existuje spousta jiných standardů čárových kódů, které jsou používány v různých oblastech jako zdravotnictví, pojišťovnictví, přeprava, výroba aj. V dnešní době je používáno více než 50 typů čárových kódů.

Čárové kódy mohou být čteny prostřednictvím laserových čtecích zařízení. Tyto zařízení svítí laserem na čárový kód a měří intenzitu odrazu. Poskytují snadné a rychlé načtení čárového kódu s takřka nulovou citlivostí na okolní osvětlení. Další alternativa čtení je pomocí digitální kamery, které také poskytují kvalitní načtení čárového kódu, ale jsou manipulačně náročnější z důvodů zajištění ostrosti obrazu při snímání.

Nicméně stále více roste zájem o přístup k čárovým kódům v terénu pomocí mobilních zařízení. V dnešní době jsou stále více rozšířenější PDA[1] a mobilní telefony. Tyto mobilní zařízení jsou většinou vybaveny digitálními kamerami a technologií umožňující přístup na internet. Mohou být dokonce použity jako čtečky čárových kódů. V kombinaci s novými službami to může znamenat revoluci při každodenních nákupech zahrnující získávání informací o produktu: přístup na hodnocení produktu, umístění podobných produktů nebo služeb a srovnání cen.

1.1 Cíl práce

Cílem této práce je navrhnout a implementovat aplikaci, která bude umožňovat čtení čárových kódů jak ze statických snímků, tak i přímé (*real-time*) čtení z digitální kamery připojené k PC. Aplikace by měla být tvořena dvěma hlavními na sebe navazujícími částmi. První část aplikace by měla zajišťovat nalezení polohy a orientace čárového kódu v obrazu. Druhá část by se měla zabývat samotným dekódováním čárového kódu.

[1] *Personal Digital Assistant* - malý kapesní počítač.

Pro *real-time* čtení čárových kódů je nezbytně nutné použít rychlé a nenáročné metody. Rychlost metod souvisí s kvalitou výsledku, tedy s úspěšností lokalizace a dekódování čárového kódu. Měl by být brán velký ohled na návrh rychlých a spolehlivých metod pro lokalizaci a dekódování čárového kódu.

Existuje několik typů čárových kódů. Zabývat se všemi by bylo nad rámec této práce, a proto bude pozornost věnována pouze čárovému kódu EAN-13 (*Europan Article Number*), který je nejvíce využíván k označování zboží.

1.2 Členění práce

Práce je členěna do sedmi kapitol:

Kapitola 1 je věnována tomuto úvodu.

Kapitola 2 pojednává o čárových kódech, kde jsou uvedeny kategorie do kterých se čárové kódy dělí a některé jejich důležité vlastnosti. Pozornost je především věnována čárovému kódu EAN-13, jehož dekódováním se práce zabývá.

V kapitole 3 jsou shrnuty možné přístupy z nastudovaných prací, které se zabývají problematikou dekódování čárového kódu. Jednak jsou zde uvedeny přístupy k samotnému dekódování, ale také i k lokalizaci čárového kódu, která s touto problematikou úzce souvisí.

Kapitola 4 se zabývá návrhem celého systému, kde je brán zřetel na nízkou výpočetní náročnost a vysokou úspěšnost dekódování.

V kapitole 5 jsou popsány implementační prostředky s jejichž pomocí bylo dosaženo výsledné aplikace. Dále je stručně popsána výsledná implementace v podobě popisu tříd a některých jejich významných metod.

Kapitola 6 je věnována testům výsledné implementace. Je zde také popsána testovací sada a nechybí ani porovnání výsledků s komerčními aplikacemi.

V závěrečné kapitole 7 je rekapitulace této práce s návrhy možného pokračování.

Kapitola 2

Čárový kód

V současné době je používáno více než 50 typů čárových kódů. Jsou totiž velmi populární v různých oblastech např. prodej, přeprava, výroba, zdravotnictví, které mají specifické požadavky na vlastnosti čárových kódů.

Tato kapitola seznamuje čtenáře s různými typy a důležitými charakteristikami čárovým kódů. Zvláště je zde věnován prostor podrobnému rozboru čárového kódu EAN-13, jehož dekódováním se celá práce zabývá.

2.1 Symboliky

Běžné čárové kódy jsou myšleny jako paralelní uspořádání čar a mezer o různých šířkách. Této definici však neodpovídají některé poštovní čárové kódy, které používají k zakódování různé výšky čar, ale jejich šířky jsou konstantní. Můžeme tedy rozeznávat [18]:

- **Šířkově modulovaný čárový kód** - paralelní uspořádání různých šířek čar a mezer.
- **Výškově modulovaný čárový kód** - paralelní uspořádání různých výšek čar.

Symbolika je termín, který je používán k popisu kódování informace do čar a mezer. Jinými slovy je to množina pravidel pro konkrétní typ čárového kódu.

Běžné čárové kódy jsou myšleny jako směsice tmavých čar a světlých mezer. Můžeme si představit symboliku, která je založena na Morseově abecedě. Tečky mohou reprezentovat úzké čáry a naopak čárky mohou reprezentovat široké čáry. Tato symbolika však není praktická, neboť každý znak je kódován různým počtem čar, což je obtížné pro dekódování. Je neefektivní, protože jen čáry mohou obsahovat smysluplné data. Použití této symboliky by mělo za následek mnoho chyb při dekódování v důsledku chyby tisku. Změna šířky, odebrání nebo přidání čáry vede k chybné interpretaci kódovaného znaku.

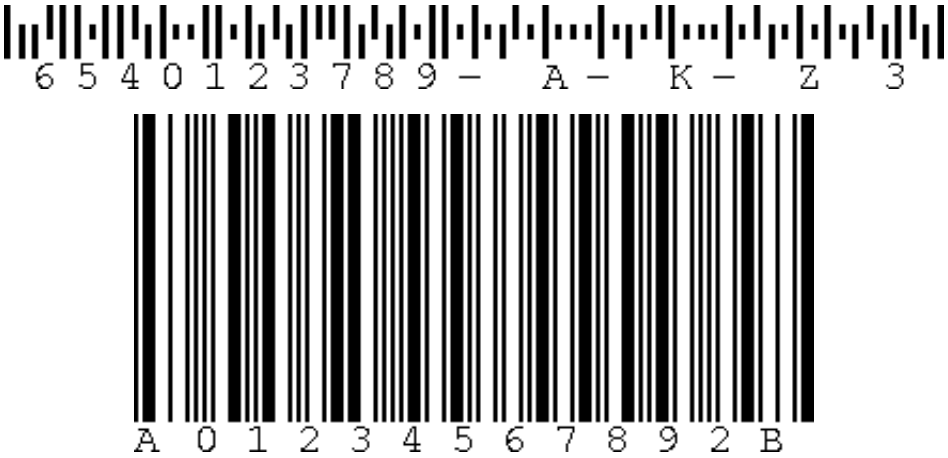
Čárové kódy nemusí nutně kódovat informaci do vzorů čar. Zde je uveden malý přehled používaných symbolik [18]:

- **Lineární symboliky** používají k zakódování dat jen jeden řádek čar a mezer. Čárový kód může být šířkově modulovaný i výškově modulovaný (viz obr. 2.1).
- **Kruhové symboliky** používají různou šířku soustředných kruhů nebo oblouků. U kódování do soustředných kruhů je možné číst čárový kód z jakéhokoliv úhlu. Ukázka kruhových symbolik je na obrázku 2.2.

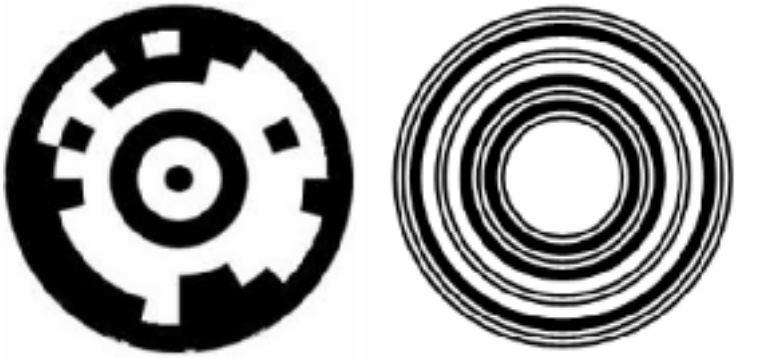
- **2-D skládané symboliky** používají více řad šířkově modulovaných čar a mezer. Každý řádek má stejnou délku a vypadá jako běžná lineární symbolika. Sousední řady se vzájemně dotýkají, nebo se mezi nimi může vyskytovat dělící čára. Ukázka takové symboliky je na obrázku 2.3.

- **2-D maticové symboliky** uchovávají informaci ve dvoudimenzionální struktuře datových buněk. Datové buňky mohou mít různé barvy (obvykle černá a bílá) a tvary (čtverce, body, polygony), které mohou být odděleny prostorem nezahrnující informace. Vyžadují sofistikované zařízení pro tisk a čtení, neboť jsou více komplikovanější než lineární nebo 2-D skládané symboliky. Poskytují mnohem vyšší hustotu dat, než u výše zmíněných symbolik. Ukázka takové symboliky je na obrázku 2.4.

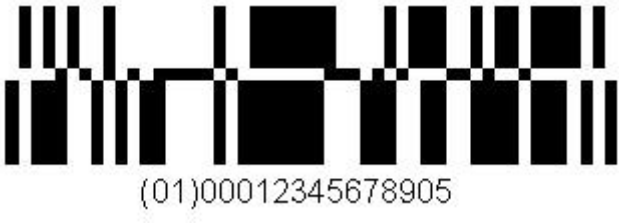
- **Kompozitní symboliky** se skládají z lineární složky a přilehlé 2-D složky (viz obr. 2.5). Lineární složka kóduje základní krátkou informaci a je navržená pro snadné a rychlé čtení obvykle laserovými čtečkami. 2-D složka obsahuje doplňující informace lineární složky. 2-D složka je obvykle čtena v jinou dobu než lineární složka a pravděpodobně i jiným zařízením jako je digitální kamera.



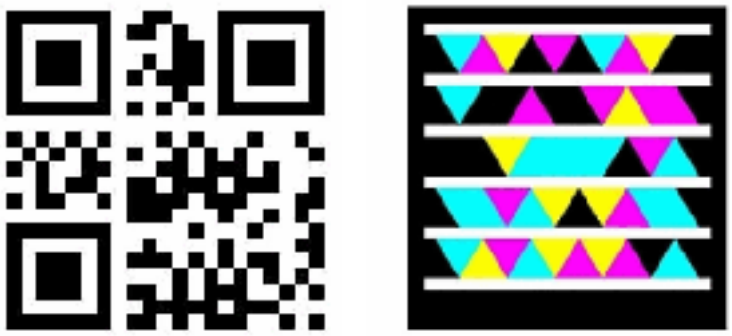
Obrázek 2.1: Lineární symboliky; Nahoře: výškově modulovaný čárový kód; Dole: šířkově modulovaný čárový kód.



Obrázek 2.2: Kruhové symboliky. Informace je kódována do soustředných oblouků nebo kruhů.



Obrázek 2.3: 2-D skládané symboliky. Více lineárních čárových kódů je poskládáno na sebe a tvoří tak jeden čárový kód.



Obrázek 2.4: 2-D maticové symboliky. Informace jsou kódovány do dvoudimenzionální struktury.

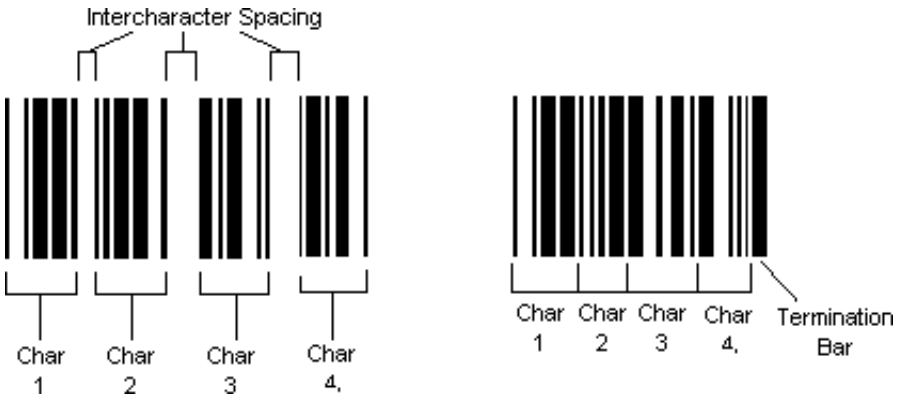
Obrázek 2.5: Kompozitní symboliky. Čárový kód je tvořen lineární složkou obsahující základní informace a 2-D maticovou složkou, která obsahuje doplňující informace lineární složky.

2.2 Vlastnosti symbolik

- **Znaková sada** - Popisuje rozsah znaků, které mohou být zakódovány danou symbolikou. Některé symboliky umožňují zakódovat pouze číslice, a proto se nazývají numerickými. Jiné mohou zakódovat alfanumerickou informaci a další dokonce umožňují zakódovat celou ASCII[1] množinu znaků.
- **Typ symboliky** - Šířkově modulované symboliky lze rozdělit do dvou hlavních kategorií, na spojité a diskrétní. U diskrétních může být každý kódovaný znak dekódován nezávisle na ostatních - začíná čarou a též končí čarou. Kódované znaky jsou od sebe odděleny mezerou, která neobsahuje žádnou informaci. Spojité symboliky

[1] *American Standard Code for Information Interchange* - kódová tabulka, která definuje znaky anglické abecedy a jiné znaky používané v informatice.

naopak neobsahují mezery oddělující kódované znaky. Konec kódovaného znaku je indikován začátkem následujícího kódovaného znaku. Z toho plyne, že spojitá symbolika umožňuje zakódovat stejná data na mnohem menší délku, než diskrétní symbolika. Diskrétní a spojité kódování znázorňuje obrázek 2.6.



Obrázek 2.6: Vlevo: diskrétní kódování; Vpravo: spojité kódování.

- **Počet šířek** - V šířkově modulovaném čárovém kódu jsou data kódovány do šířek čar a mezer. Jednotlivé šířky jsou rovny násobkům základní (minimální) šířky čáry. Existují dva základní typy čárových kódů: ty které používají pouze dvě šířky (tenká a široká) a ty, jenž používají více šířek. Většina symbolik používajících větší množství šířek je modulárních tj. délka symbolu je předem dělena do daného počtu modulů a téměř všechny jsou spojitého typu.

- **Pevná nebo proměnná délka** - Některé symboliky mohou zakódovat informaci o libovolné délce a jiné naopak naopak dovolují zakódovat pouze pevně dané množství znaků.

- **Hustota** - Lineární symboliky se liší množstvím zakódovaných dat na danou délku. Hustota dat je počítána jako poměr zakódovaných znaků k délce čárového kódu, která vyjádřena v počtu základních (minimálních) šířek čar čárového kódu.

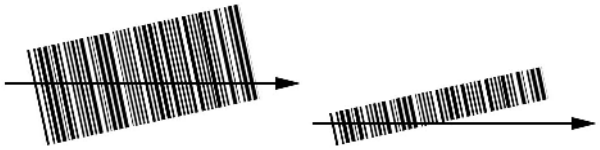
- **Vlastní kontrola** - Symboliky s vlastní kontrolou jsou takové, u kterých např. tisková vada nezpůsobí při dekódování záměnu znaku, který by vedl k validnímu kódu.

- **Start symbol, stop symbol** - Start symbol je speciální vzor čar, který je umístěn na začátku čárového kódu. Slouží čtecímu zařízení k indikaci počátku čárového kódu. Naopak je tomu u stop symbolu. Start a stop symbol u některých symbolik určuje i směr čtení.

- **Hledaný vzor** - 2-D maticové symboliky nepoužívají čáry a mezery k reprezentaci kódovaných dat. Mají strukturu datových buněk, kterou může být obtížné nalézt v obrazu. Pro snadné zjištění polohy a orientace čárového kódu jsou do těchto symbolik zahrnuty jedinečné vzory nebo tvary.

- **Kontrolní symbol** - Nachází se na určité pozici daného čárového kódu. Jeho hodnota je určena dle nějakého matematického vztahu. Je používán k validaci dat, která jsou dekódována. Pokud kontrolní symbol může nabývat pouze čísel, pak je nazýván kontrolní číslicí.

- **Oprava chyb** - V těchto symbolikách jsou redundantní data, která umožňují při dekódování opravu chyb.
- **Obousměrnost** - U obousměrných symbolik nezáleží na tom, zda čtecí zařízení bude číst čárový kód zleva doprava nebo naopak. Výsledek dekódovaných dat bude vždy stejný.
- **Klidová zóna** - Čárové kódy nejsou vždy tisknuty na velmi velkém bílém pozadí, ale jsou často blízko jiné tisknuté grafiky. Klidová zóna fyzicky odděluje čárový kód od ostatních údajů resp. grafiky na objektu, což umožňuje čtecím zařízením snadno nalézt čárový kód. Velikost klidové zóny se liší v závislosti na použité symbolice.
- **Kódování a dekódování** - Čárový kód je nejčastěji tvořen sekvencí vzorů jednotlivých zakódovaných znaků. Široké nebo úzké čáry a mezery mohou binárně kódovat libovolné znaky. Tyto symboliky jsou snadno kódovatelné a dekódovatelné. Moderní a vysoce efektivní symboliky se často odchylují od původních ideí, kde jeden znak je reprezentován jedním kódovaným vzorem. Místo toho jsou data častěji komprimována, přeformátována a distribuována přes několik kódovaných znaků.
- **Výška** - Výška čárového kódu ovlivňuje rychlost dekódování. Čárové kódy s malou výškou omezují čtecí zařízení v rozsahu úhlů při kterém lze načíst celý čárový kód. Takové hledání správného čtecího úhlu u čárových kódu s malou výškou je časově náročnější než u vyšších čárových kódů (viz obr. 2.7).



Obrázek 2.7: Problém čtení čárových kódů s malou výškou.

2.3 Čárový kód EAN-13

Zkratka EAN znamená *European Article Number*. Kódy EAN-13 jsou používány po celém světě k označování jednotlivých druhů zboží. Tento čárový kód obsahuje třináct číslic. První dvě nebo tři číslice slouží jako identifikátor země. Následujících devět nebo deset číslic (záleží na identifikátoru země) označuje kód výrobce a kód produktu.

Jsou také používány kódy EAN-8, které jsou výhradně aplikovány na menší položky u nichž by byl problém umístit čárový kód EAN-13.

Tyto čárové kódy jsou složeny z čar a mezer různých šířek. Nejmenší šířka čáry či mezery se nazývá modul. Jednomu modulu čáry odpovídá logická 1 a naopak modulu mezery odpovídá logická 0.

Čárový kód EAN-13 kóduje pouze číslice a každá z nich je kódována čtyřmi šířkami. Jsou to dvě šířky čar a dvě šířky mezer, které mohou mít velikost jednoho, dvou, tří nebo čtyř modulů. Celková velikost kódované číslice činí sedm modulů.

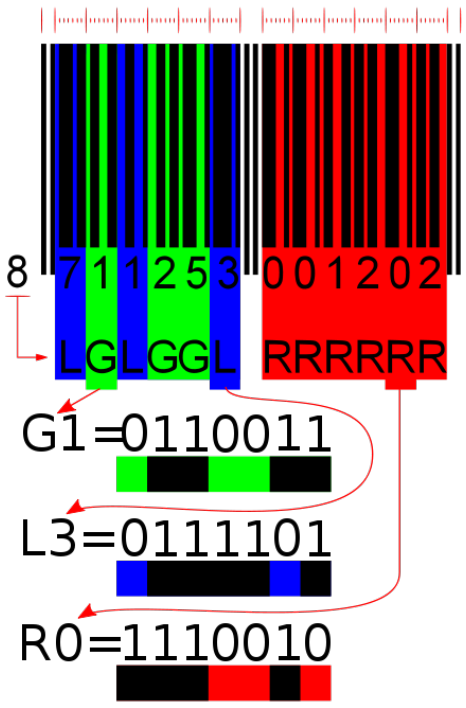
V čárovém kódu se také vyskytují dva speciální symboly. Je to start a stop symbol (101), který vymezuje oblast čárového kódu a středový symbol (01010), který rozděluje čárový kód na dvě skupiny po šesti zakódovaných číslicích. V první skupině lze každou číslici kódovat pomocí dvou různých kódovacích sad L a G. Druhá skupina se kóduje jen

pomocí kódovací sady R. Použití dvou různých kódovacích sad v první skupině šesti číslic má speciální význam. Podle vzoru dvou použitých sad je zakódována počáteční číslice čárového kódu (viz tab. 2.1). Pro názornost je systém kódování uveden na obrázku[2] 2.8.

EAN-13 využívá samodetekující kód k ověření správnosti. Kontrola správnosti probíhá sečtením číslic, které se vyskytují na lichých pozicích. K tomuto součtu se přičte trojnásobek součtu číslic na sudých pozicích a výsledná suma pak musí být dělitelná deseti. Pokud tomu tak není, nejedná o validní kód.

Tabulka 2.1: Kódovací sady čárového kódu EAN-13 včetně vzorů použitých kódovacích sad pro zakódování počáteční číslice.

Číslice	Kód. sada L	Kód. sada G	Kód. sada R	Vzor
0	0001101	0100111	1110010	LLLLLL
1	0011001	0110011	1100110	LLGLGG
2	0010011	0011011	1101100	LLGGLG
3	0111101	0100001	1000010	LLGGGL
4	0100011	0011101	1011100	LGLLGG
5	0110001	0111001	1001110	LGGLLG
6	0101111	0000101	1010000	LGGGLL
7	0111011	0010001	1000100	LGLGLG
8	0110111	0001001	1001000	LGLGGL
9	0001011	0010111	1110100	LGGLGL



Obrázek 2.8: Kódování čárového kódu EAN-13.

[2]Obrázek pochází z http://en.wikipedia.org/wiki/European_Article_Number

Kapitola 3

Existující přístupy

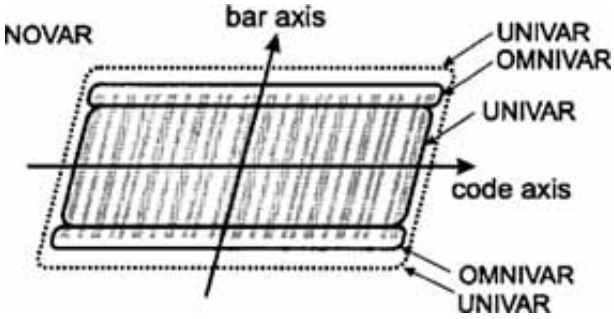
Problém dekódování čárového kódu v obrazu je běžně rozdělován na dvě hlavní části. Jedná se o lokalizaci a dekódování. Lokalizace zajišťuje nalezení oblasti s čárovým kódem. Dekódování pak z této nalezené oblasti získá informace, které byly vloženy do čar čárového kódu. Toto rozdělení zajišťuje snížení výpočetní náročnosti celého systému.

3.1 Lokalizace čárového kódu

Běžný lineární čárový kód je tvořen paralelní posloupností čar. Poskytuje tedy vhodnou homogenní texturu, která může být snadno lokalizována.

Většina metod [1, 2, 5, 10, 20, 25] využívá k takové lokalizaci gradientů, které nesou informaci o směru a síle hrany v obrazu. Tyto metody používají Sobelův nebo Robinsonův operátor k vytvoření gradientní mapy zkoumaného obrazu. V této gradientní mapě jsou zkoumány směrové charakteristiky jednotlivých gradientů, kde oblast čárového kódu je charakteristická gradienty se stejnou orientací. Tato lokalizace může být prováděna prahováním obrazu po blocích, kde optimální gradientní práh je zvolen na základě histogramu gradientní mapy [10]. To umožňuje oddělit čárový kód od okolí. V [20] je gradientní mapa využita pouze ke zjištění orientace čárového kódu. Dle této orientace je celý obraz otočen, aby se čárový kód nacházel ve vodorovné poloze. Poté jsou hranice čárového kódu hledány v průběhu intenzit jasu (odstínů šedi) obrazových bodů několika čtených řádků obrazu. V [1] jsou vytvářeny dva binární obrazy pro horizontální a vertikální orientace gradientů. Do binárních obrazů nejsou zahrnuty gradienty, které mají malou velikost nebo diagonální orientaci. Poté jsou na těchto obrazech vykonávány morfologické operace a v konečné fázi je provedena analýza těchto obrazů k nalezení polohy čárového kódu. Podobného principu je využito v [8, 25], kde jsou vytvářeny místo dvou, čtyři binární obrazy pro gradienty s orientacemi 0°, 45°, 90° a 135°. S komplexním využitím gradientů přichází [5]. Gradienty jsou použity ke klasifikaci obrazových segmentů do tří kategorií (viz obr. 3.1): UNIVAR (*undirectional variation*), OMNIVAR (*omnidirectionally variation*) a NOVAR (*no variation*). Analýzou obrazových segmentů spadajících do výše zmíněných kategorií je provedena lokalizace čárového kódu.

Mnoho segmentačních úloh je prováděno pomocí metody Houghovy transformace, která hledá parametrický popis objektů v obrazu. Používá se k detekci jednoduchých tvarů jako jsou přímky, kružnice, elipsy atd. V [17, 28] je metoda Houghovy transformace využita k lokalizaci čárového kódu. Taková lokalizace je velmi odolná proti šumu v obrazu, ale pro čárové kódy, které jsou na nerovném povrchu má jisté problémy.



Obrázek 3.1: Rozložení UNIVAR, OMNIVAR a NOVAR v čárovém kódu [5].

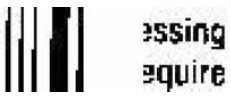
K detekci hran jsou také používány Gaborovi filtry, které jsou použity v [11]. Obraz je nejprve filtrován osmi Gaborovými filtry v orientacích 0°; 22,5°; 45°; 67,5°; 90°; 112,5°; 135° a 157,5°. Z těchto filtrů jsou získány obrazové rysy, které jsou využity ke klasifikaci oblasti čárového kódu pomocí neuronové sítě. Tato metoda je robustní na orientaci a zakřivení čárového kódu, ale na druhou stranu má vyšší výpočetní nároky.

Další metody k lokalizaci používají morfologické operace [4, 7]. Např. v [7] je obraz nejprve binarizován, poté je rozdělen na jednotlivé nepřekrývající se bloky (viz obr. 3.2). Na každém bloku jsou provedeny morfologické operace, pomocí kterých se dosáhne „skeletonizace“ (viz obr. 3.3). Z těchto bloků jsou dále extrahovány jednotlivé spojené složky (viz obr. 3.4) a následně dle jejich vzájemné rovnoběžnosti je rozhodnuto, zda daný blok je, či není součástí čárového kódu.

Zajímavá metoda lokalizace je prezentována v [15]. Tato metoda využívá několika skupin paralelně čtených řádků v obrazu, které mají orientace 0°, 45°, 90° a 135°. Každý takový řádek představuje profil jasové intenzity obrazových bodů. Profily intenzit jsou podrobeny hledání hranic čárového kódu, dle kterých je vypočtena poloha a orientace čárového kódu. Tato metoda je efektivní, protože může nalézt čárový kód z malého množství čtených řádků. U některých nekvalitních snímků však vyžaduje trochu více času na předzpracování. Podobný způsob lokalizace lze také nalézt i v [9].

Existují i takové metody, které si lokalizaci značně zjednodušují předpokladem, že část čárového kódu je umístěna ve středu obrazu. V [16] probíhá vymezení oblasti čárového kódu spirálovitým hledáním čar od středu obrazu.

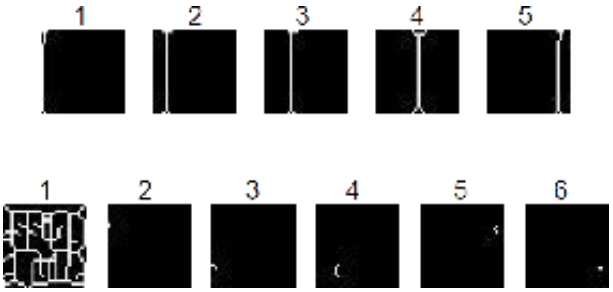
Další neobvyklé metody lokalizace využívají frekvenční oblast obrazu, a to použitím vlnkové transformace [27] nebo diskrétní kosinové transformace [24]. Diskrétní kosinová transformace (DCT) je také využívána při kompresy JPEG[1] obrazových souborů. Faktem je, že JPEG kodeky jsou často implementovány v hardwaru kamery a mohou tak poskytovat rychlý přístup do DCT domény obrazu s čárovým kódem.



Obrázek 3.2: Dva binarizované bloky zkoumaného obrazu [7].

[1] *JPEG File Interchange Format* - standardní metoda ztrátové komprese používané pro ukládání počítačových obrázků ve fotorealistické kvalitě

Obrázek 3.3: Skeletonizované bloky [7].



Obrázek 3.4: Extrakce spojených složek ze skeletonizovaných bloků [7].

3.2 Dekódování čárového kódu

Po lokalizaci nastává proces dekódování čárového kódu. Informace je u běžných lineárních čárových kódů zašifrována do paralelní posloupnosti čar a mezer o různých šířkách a je tedy nutné získat posloupnost těchto šířek.

To je obvykle prováděno binarizací[2] intenzit jasu obrazových bodů, které jsou postupně čteny skrze celý čárový kód. Jednoduchá binarizace takového sledu obrazových bodů je prováděna v [7], kde je hodnota prahu zvolena na základě průměru všech přečtených jasových intenzit obrazových bodů. V obrazu s čárovým kódem se mohou též vyskytovat za špatných světelných podmínek tmavé a světlé oblasti, nebo také šum způsobený nekvalitním snímacím zařízením. Z tohoto důvodu některé metody provádí mnohonásobnou binarizaci s různou prahovou hodnotou [20], nebo je binarizováno více sledů obrazových bodů s odlišnými hodnotami prahu [3]. Dalším možným řešením tohoto problému je volit práh dynamicky podle lokálních minim a maxim v průběhu jasových intenzit obrazových bodů [26].

Informaci o šířkách lze také získat z pozic jednotlivých hran čárového kódu. Získání těchto pozic probíhá hledáním průchodů nulou ve druhé derivaci profilu jasových intenzit obrazových bodů [2, 21, 27]. Vlivem šumu však dochází k detekci falešných hran, které se většinou vykazují malými lokálními extrémy, které lze z výsledné množiny hran odfiltrovat.

Přesnost výše zmíněných metod klesá se zvyšujícím se rozmazáním snímků, které mohou být způsobeny pohybem při snímání nebo špatným zaostřením. Metody, které jsou odolnější vůči rozmazání vychází z důkladné analýzy profilu jasových intenzit. Především se jedná o analýzu lokálních extrémů, jenž nesou informaci o střední poloze čar a mezer [12, 13].

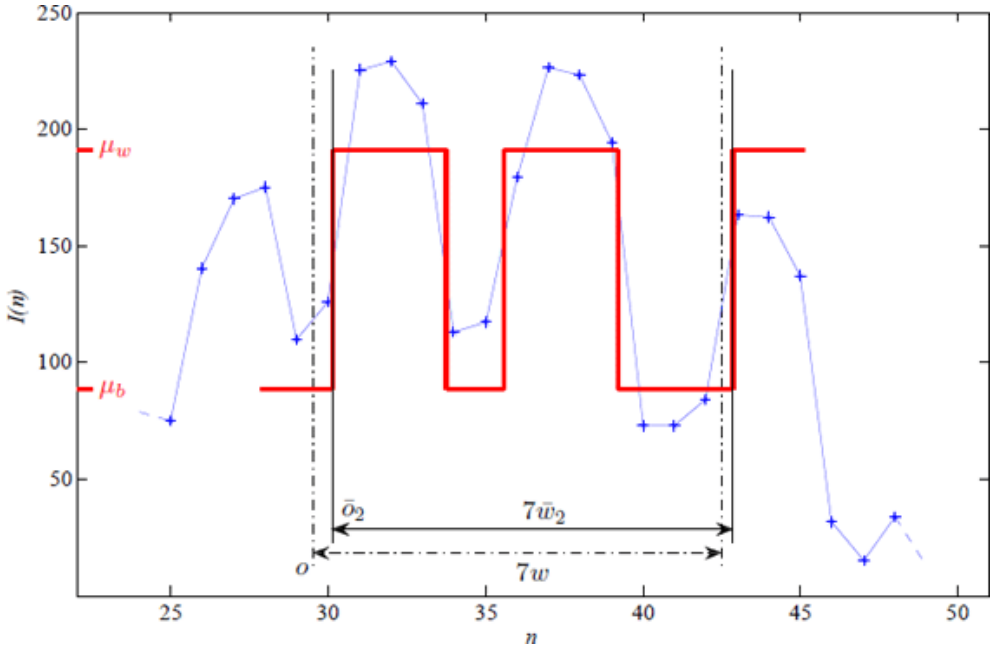
Dekódovaní na základě šířek čar a mezer může být prováděno statistickým rozpoznáváním [27], nebo také pomocí neuronové sítě [14, 28]. V [23] je představen „Bayesův model“, pomocí kterého je prováděno dekódování na základě pravděpodobnosti a znalosti geometrie čárového kódu (povolené konfigurace čar a mezer). Autoři tvrdí, že tento algoritmus je odolný vůči geometrickému zkreslení, šumu i chybějícím hranám čárového kódu.

V [10] autoři představili nový přístup k dekódování čárového kódu, který neprovádí žádnou binarizaci ani hledání hran. Pro každý kódovaný znak je vytvořen model ideálního

[2]Kategorizace hodnot do dvou skupin - např. černá a bílá.

průběhu jasové intenzity. Tyto modely slouží k porovnávání s reálným průběhem intenzit jasu. Hledají se tedy takové modely, které nejvíce odpovídají dílčím úsekům reálného průběhu jasových intenzit. Ukázka porovnání jednoho modelu je na obrázku 3.5.

Zcela odlišný způsob dekódování je uveden v [1]. Vychází z toho, že téměř všechny lineární čárové kódy jsou vespod doplněny nekódovanou informací v podobě posloupnosti znaků, které jsou čitelné pro člověka. Tato práce se tedy zabývá lokalizací a dekódováním číslic, které se nachází pod čárovým kódem. Klasifikace číslic je prováděna na základě Bayesových pravidel.



Obrázek 3.5: Porovnání jednoho modelu (červená čára) s průběhem jasové intenzity čárového kódu (modrá čára) [10].

Kapitola 4

Návrh

Tato kapitola se zabývá návrhem systému pro dekódování čárového kódu v reálném čase. Celý návrh je zaměřen na lineární čárový kód EAN-13 a je rozdělen do třech funkčních celků, které na sebe navzájem navazují. Tyto tři funkční celky tvoří:

1. **Lokalizace** - Zabývá se nalezením přibližné polohy a orientace čárového kódu v obraze.
2. **Binarizace** - Dle informací o poloze a orientaci čárového kódu dochází k přečtení průběhu jasových intenzit skrz čárový kód. Poté je průběh jasových intenzit převeden binarizací na „posloupnost čar a mezer“.
3. **Dekódování** - V „posloupnosti čar a mezer“ jsou nejprve nelezeny hranice čárového kódu a následně je provedeno dekódování číselné informace, která je obsažena v čárách a mezerách. V poslední fázi je ověřena správnost dekódovaného řetězce dle kontrolní číslice.

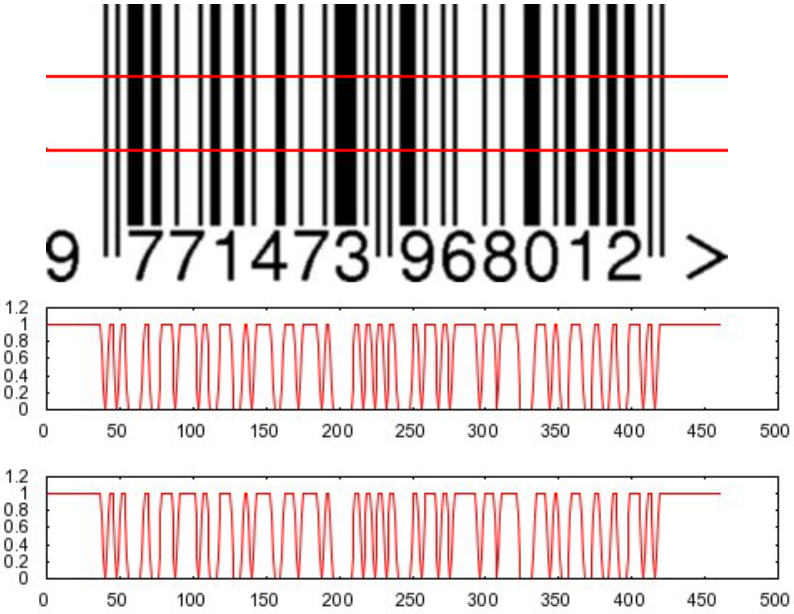
4.1 Lokalizace

Čárový kód EAN-13, kterým se zabývám v této práci je tvořen posloupností několika navzájem rovnoběžných čar a mezer. Této skutečnosti využívá mnou navržená metoda lokalizace ke které mě inspirovaly poznatky z [15]. Spočívá v analýze resp. v hledání podobností mezi profily jasových intenzit.

Na obrázku 4.1 jsou znázorněny dva profily jasových intenzit, které byly získány přečtením dvou řádků obrazu směřujících kolmo na čárový kód. Na těchto profilech lze vypozorovat vzájemnou podobnost a je tedy možné tvrdit, že v oblasti mezi přečtenými řádky, kde nastává přibližná shoda profilů jasových intenzit se nachází část čárového kódu.

Na obrázku 4.2 je čárový kód natočen o 20°. V tomto případě však nedochází ke čtení řádků směřujících kolmo na čárový kód, nicméně lze i zde vypozorovat shodné oblasti profilů, které jsou navzájem mírně posunuty. Posun profilů má velký význam k určení orientace, který je dále podrobněji rozebrán. Je nutno podotknout, že v tomto případě nedochází k přečtení jasových intenzit celého čárového kódu, což znesnadňuje lokalizaci hranic. To způsobuje fyzická výška čárového kódu a také jeho orientace v obraze. Mírné natočení čárového kódu je možné řešit snižováním kroku mezi čtenými řádky. Větší natočení je nutné již řešit čtením posloupnosti jasových intenzit v několika směrech. Z důvodů výpočetní náročnosti je prováděno čtení pouze ve čtyřech směrech v orientacích 0°, 45°, 90° a 135°. Každý směr tedy pokrývá rozsah natočení čárového kódu o ±22,5°.

Cílem návrhu této lokalizace není určit přesnou oblast čárového kódu, ale pouze přibližnou polohu a orientaci. Určení hranic a užitečných čar a mezer čárového kódu je záležitostí dekódovacího procesu.



Obrázek 4.1: Přečtení dvou řádků čárového kódu s příslušnými profily jasových intenzit.

Obrázek 4.2: Přečtení dvou řádků čárového kódu s příslušnými profily jasových intenzit.

4.1.1 Nalezení podobností

Lokalizace je založena na nalezení podobností v profilech jasových intenzit čtených řádků. Profily jasových intenzit lze také chápat jako průběhy diskrétních signalů. V oblasti zpracování signálů se k měření podobnosti dvou signálů používá korelace. Korelace dvou diskrétních signálů f a g je dána vztahem 4.1. Z uvedeného vztahu je patrné, že korelace je docela náročná, neboť vyžaduje velké množství součinů v závislosti na délce signálů.

$$(f \star g) \stackrel{\text{def}}{=} \sum_{m=-\infty}^{\infty} f^*[m]g[n+m] \tag{4.1}$$

Nalezení vzájemných podobností mezi signály jsem se tedy rozhodl řešit porovnáváním význačných bodů v profilech jasových intenzit. Význačný bod jsem definoval jako posloupnost pěti lokálních extrémů (minim a maxim), kde rozdíly v jasových intenzitách mezi sousedními extrémy resp. rozdíly mezi maximi a minimi jsou větší než daný práh P[1]. Význačný bod lze tedy zapsat jako vektor $B_i = (e_{i-2}, e_{i-1}, e_i, e_{i+1}, e_{i+2})$, který splňuje podmínky:

$$\begin{aligned} |e_{i-2} - e_{i-1}| &> P \\ |e_{i-1} - e_i| &> P \\ |e_i - e_{i+1}| &> P \\ |e_{i+1} - e_{i+2}| &> P \end{aligned}$$

Určení podobnosti mezi význačnými body dvou profilů jasových intenzit probíhá tak, že každý význačný bod prvního profilu je porovnáván se všemi význačnými body druhého profilu, jenž spadají do prohledávané oblasti. Pro každou dvojici porovnávaných význačných bodů je vypočtena míra diference d dle vztahu 4.2, kde e_{ai} a e_{bi} jsou jednotlivé složky význačných bodů a a b. Nejvyšší podobnost dvou význačných bodu je dána nejnižší hodnotou diference.

$$d(a,b) = \sum_{i=1}^{5} |e_{ai} - e_{bi}| \tag{4.2}$$

4.1.2 Určení polohy a orientace

Předpokladem pro určení polohy a orientace čárového kódu v obrazu jsou nalezené podobné význačné body mezi sousedními jasovými profily, jenž spadají do stejné skupiny čtení. Jak jsem již zmiňoval, tyto skupiny jasových profilů jsou čtyři, a to v orientacích čtení 0°, 45°, 90° a 135°. Výpočet polohy a orientace čárového kodů je proveden pouze z jedné skupiny a podstatné je tedy nejprve zvolit kandidátní skupinu. Obecně je to taková, při které čtení jasových intenzit probíhalo v kolmém směru na čárový kód, což se také projevuje velkou hustotou podobných význačných bodů.

Na obrázku 4.4 je zobrazen čárový kód, který je v obraze natočen o 20° s překrývající mřížkou. Tato mřížka představuje čtené řádky jasových intenzit v horizontálním, vertikálním a dvou diagonálních směrech. Obrázky 4.5 až 4.8 znázorňují přečtené řádky jasových intenzit poskládané pod sebou ve výše jmenovaných směrech čtení. Na obrázcích jsou také

[1] Empiricky bylo zjištěno že tento práh je roven hodnotě 40.

vykresleny červené body, které odpovídají význačným bodům se vzájemnými podobnostmi. Čárový kód je v obraze natočen o 20°, což je na hranici mezi čtenými řádky v orientacích 0° a 45°. To se také projevuje zvýšeným počtem podobných význačných bodů v těchto čtených orientacích (viz obr. 4.5 a 4.6).

V nalezení vhodné skupiny (ze které bude proveden výpočet polohy a orientace čárového kódu) jasových profilů hraje významnou roli offset, neboli posun podobných význačných bodů mezi sousedními řádky. Pro každou skupinu je vytvořen histogram offsetů, ve kterém je hledána nejvyšší četnost. Je nutno podotknout, že z hlediska diskretizace není offset vždy stabilní, a proto je žádoucí počítat s tolerancí ± 1 jednotka offsetu. O výsledné skupině je tedy rozhodnuto dle maximální sumy četností tří po sobě jdoucích offsetů v histogramu.

Přibližnou polohu je pak možné určit nalezením nejdelší souvislé posloupnosti na sebe navazujících podobných význačných bodů. To vychází ze skutečnosti, že čárový kód je tvořen z dlouhých rovnoběžných čar. Vyhledání nejdelší souvislé posloupnosti probíhá ve struktuře dvourozměrného pole, kde buňky na jejichž indexech se nachází význačný bod obsahují hodnotu offsetu ze které lze dopočítat index podobného význačného bodu ležícího na následujícím řádku. Ostatní buňky, na jejichž indexech se nenachází žádný význačný bod mají nedefinovanou hodnotu. Na obrázku 4.3 je znázorněna ukázka části takové struktury.

Poloha a orientace je pak určena ze souřadnic počátečního a koncového bodu nejdelší nalezené souvislé posloupnosti. Poloha čárového kódu je reprezentována souřadnicí bodu B v obraze, kterou je možné získat ze vztahu 4.3, kde index p značí x-ovou a y-ovou souřadnici počátečního bodu a analogicky index k označuje x-ovou a y-ovou souřadnici koncového bodu. Orientaci α lze spočíst ze vztahu 4.4 za předpokladu, že $x_p \neq x_k$.

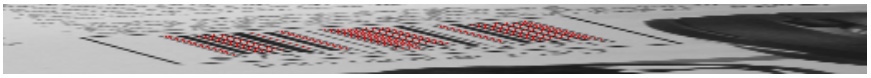
$$B = \left[\frac{x_p + x_k}{2}, \frac{y_p + y_k}{2} \right] \tag{4.3}$$

$$\alpha = \arctan\left(\frac{y_p - y_k}{x_p - x_k} \right) \tag{4.4}$$

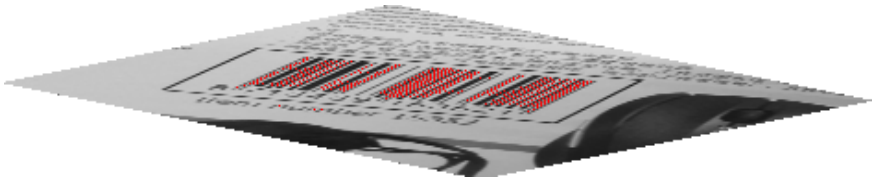
-	-	-	**1**	-	-	-	-	-	-	⋯
-	-	-	-	**0**	-	-	-	-1	-	⋯
-	1	-	-	**1**	-	-	0	-	-	⋯
-	-	1	-	-	**0**	-	-	-	-	⋯
-	-	-	1	-	**-**	-	-	-	-	⋯
-	-	-	-	-	-	-	-	-	-1	⋯
-	-	-	-	-	-	-	-	-	-	⋯
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱

Obrázek 4.3: Ukázka struktury pro vyhledání nejdelší souvislé posloupnosti podobných význačných bodů ve které je nejdelší souvislá posloupnost tvořena posloupností hodnot 1010.

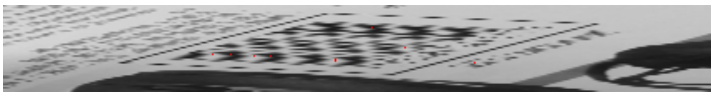
Obrázek 4.4: Obraz s čárovým kódem se znázorněnými řádky čtení.



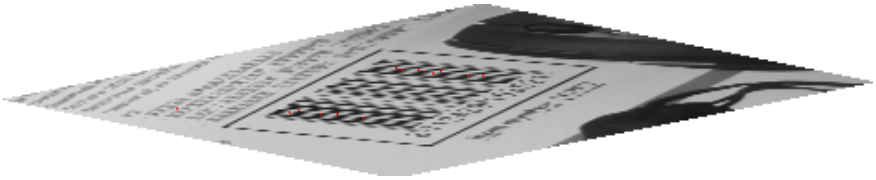
Obrázek 4.5: Podobnosti v horizontálním směru (0°).



Obrázek 4.6: Podobnosti v diagonálním směru (45°).



Obrázek 4.7: Podobnosti ve vertikálním směru (90°).



Obrázek 4.8: Podobnosti v diagonálním směru (135°).

4.2 Binarizace

Po nalezení přibližné polohy a orientace čárového kódu v obraze nastává proces binarizace. Tím lze získat z profilu jasových intenzit střídající se sledy dvou hodnot, jejichž délky (sledů) jsou nezbytné k dekódování.

Nejprve je však nutné dostat profil jasových intenzit z oblasti čárového kódu. Z přibližné polohy a orientace jsou vypočteny souřadnice počátečního a koncového bodu čtení, které mají tu vlastnost, že leží na některé z hranic obrazu (viz obr. 4.9). To umožňuje přečíst maximální délku.

Ke čtení jasových intenzit jednotlivých pixelů obrazu je použit Bresenhamův algoritmus, který je běžně používán k rasterizaci úsečky. Je velmi efektivní, neboť k určení souřadnic bodů úsečky v obrazu využívá pouze celočíselnou aritmetiku. Pro vykreslení úsečky, jejíž směrnice je kladná a menší než jedna (viz obr. 4.10) je postup Bresenhamova algoritmu následující [19]:

1. Z koncových bodů $[x_1, y_1]$ a $[x_2, y_2]$ urči konstanty $k_1 = 2\Delta y$ a $k_2 = 2(\Delta y - \Delta x)$
2. Inicializuj rozhodovací člen p na hodnotu $2\Delta y - \Delta x$
3. Inicializuj $[x, y]$ jako $[x_1, y_1]$
4. Vykresli bod $[x, y]$
5. Dokud je $x \leq x_2$, opakuj:
   (a) $x = x + 1$
   (b) Je-li p kladné, pak $y = y + 1$ a $p = p + k_2$
   (c) Není-li p kladné, pak $p = p + k_1$
   (d) Vykresli bod $[x, y]$

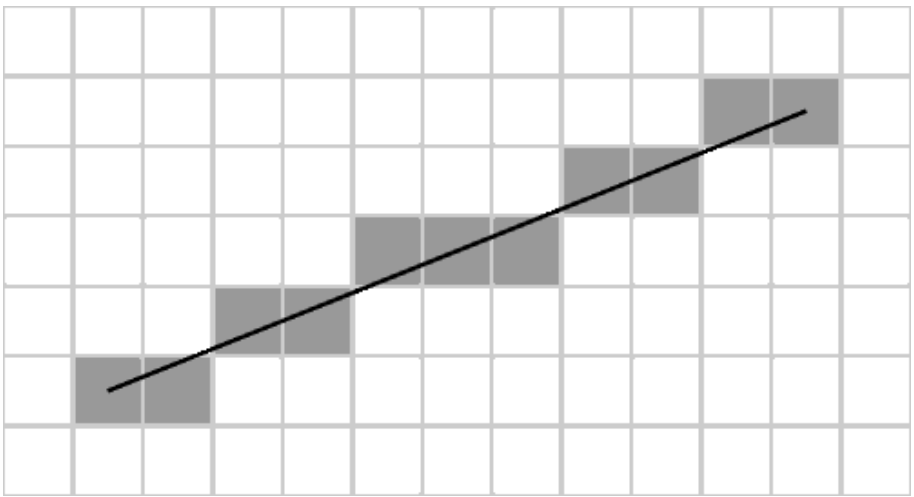
Pro vykreslení libovolných úseček je nutné v inicializační fázi algoritmu rozhodnout, která osa je řídící a podle toho zaslat úsečku do jedné ze dvou samostatných, analogicky vytvořených částí algoritmu.

Další krokem je provedení samotné binarizace průběhu jasových intenzit. Pozorováním bylo zjištěno, že některé průběhy jasových intenzit příliš kolísají, resp. diference jasových intenzit mezi potenciálními čarami a mezerami mají příliš velký rozptyl, který je závislý především na šířkách čar a mezer. Z tohoto důvodu je nemožné provádět binarizaci na základě globálního prahu, neboť by tím byly ztraceny některé důležité informace o čarách, či mezerách kódu.

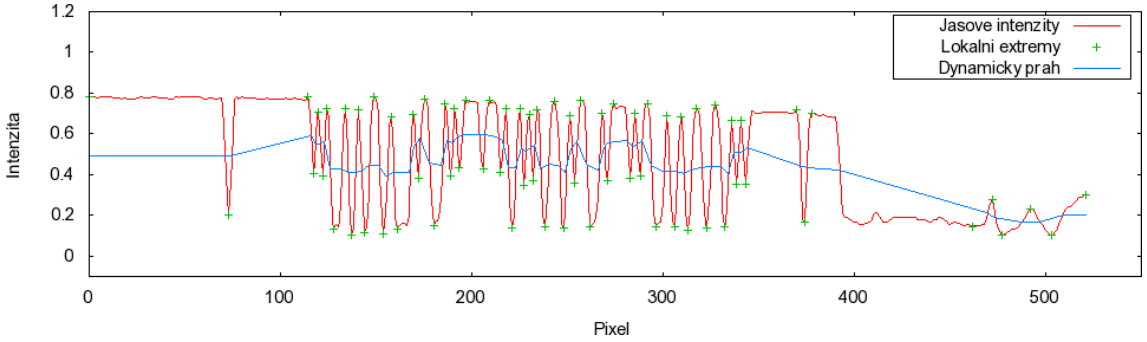
Volba padla na binarizaci prováděnou dynamickým prahováním. Jak již sám název napovídá, práh je volen dynamicky, a to podle lokálního průběhu jasových intenzit. K výpočtu dynamického prahu je nutné znát lokální extrémy. Ovšem v nevyhlazeném průběhu jasových intenzit se může nacházet příliš velké množství nežádoucích lokálních extrémů způsobených především šumem v obraze. Proto jsou vybírány pouze takové lokální extrémy, jež splňují podmínku, že rozdíl jasových intenzit aktuálního a předchozího nalezeného extrému je vyšší než hodnota 40 (bylo experimentálně zjištěno). Dynamický práh je pak volen právě mezi těmito nalezenými lokálními extrémy.

Ukázka průběhu dynamického prahu včetně lokálních extrémů a jasových intenzit je na obrázku 4.11 a odpovídající binarizace je znázorněna na obrázku 4.12.

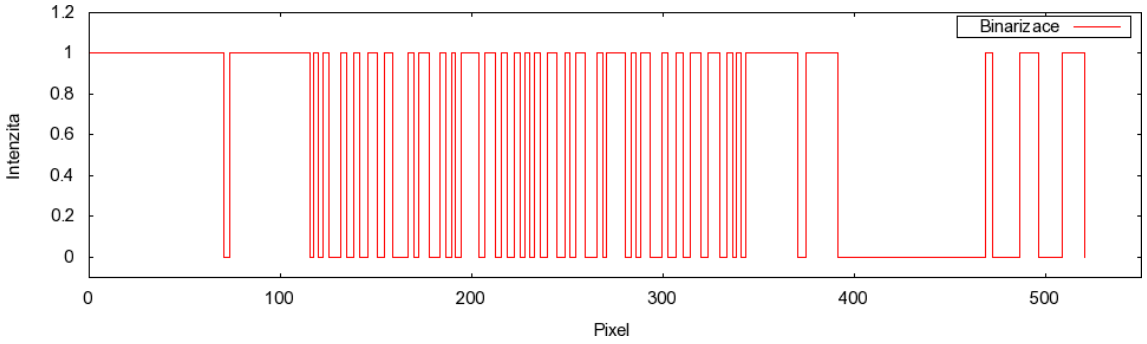
Obrázek 4.9: Nalezená poloha čárového kódu lokalizačním algoritmem (zelená čára) a získaný průběh jasových intenzit bodů potřebný k dekódování (červená čára).



Obrázek 4.10: Rasterizace úsečky.



Obrázek 4.11: Průběh jasových intenzit s vyznačenými lokálními extrémy a dynamickým prahem.



Obrázek 4.12: Binarizovaný průběh jasových intenzit.

4.3 Dekódování

V předchozím kroku byla provedena binarizace průběhu jasových intenzit, ve kterém by se měly někde nacházet čáry a mezery, jenž jsou důležité pro dekódování. Na binarizovaném průběhu jasu, který prezentuje jednotlivé šířky čar a mezer je možné provést dekódování, ale je však nutné nejprve v binarizovaném sledu čar a mezer vytyčit hranice čárového kódu.

4.3.1 Nalezení hranic

Hledání hranic čárového kódu EAN-13 vychází z následujících faktů:

- Šířky čar a mezer se v čárovém kódu vyskytují pouze v 1, 2, 3 a 4 násobku základní šířky.
- Hranice jsou tvořeny sekvencí čára-mezera-čára o velikostech základní šířky.
- Čárový kód je obklopen klidovou zónou, která je větší než čtyřnásobek základní šířky.

Pokud je nalezena sekvence čára-mezera-čára, kde vzájemné šířky jsou si podobné nebo stejné, je možné, že se jedná o hranici čárového kódu. Ovšem tato sekvence podobných šířek se může vyskytovat i uvnitř čárového kódu. Pokud za touto sekvencí následuje mezera, která je více než čtyřnásobkem průměrné šířky v sekvenci, pak se jedná o klidovou zónu a můžeme tedy s jistotou prohlásit, že sekvence je hranicí čárového kódu.

V binarizovaném průběhu je však nutné počítat s jistou nepřesností. Proto je nalezení hranic podmíněno vztahem 4.5, kde šířka potencionální klidové zóny w_{i+3} musí být více

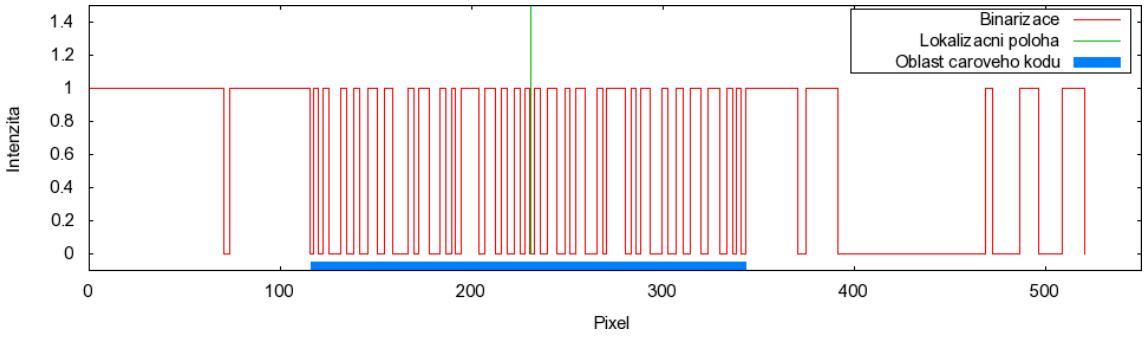
než šestinásobek průměru šířek w_i, w_{i+1}, w_{i+2} potenciální hranice. Tento vztah je možné následně upravit na 4.6, kde symbol $\ll$ reprezentuje bitový posun vlevo, což vede k vyšší efektivitě algoritmu.

$$6\frac{(w_i + w_{i+1} + w_{i+2})}{3} < w_{i+3} \tag{4.5}$$

$$2(w_i + w_{i+1} + w_{i+2}) < w_{i+3}$$

$$((w_i + w_{i+1} + w_{i+2}) \ll 1) < w_{i+3} \tag{4.6}$$

Může se stát, že podmínka 4.6 bude splněna i mimo oblast čárového kódu, což samozřejmě vede k chybnému nalezení hranic. Proto je do hledání hranic zahrnuta informace o poloze čárového kódu, která byla získána z fáze lokalizace. Na obrázku 4.9 je vyznačena úsečkou sekvence čtených pixelů, která se kříží s lokalizační úsečkou. Právě tento bod křížení je promítnut do binarizovaného průběhu na obrázku 4.13. Od tohoto bodu probíhá vymezení levé hranice postupným čtením binarizovaných šířek vlevo, dokud nebude splněna podmínka 4.6 a analogicky čtením vpravo pro nalezení pravé hranice.



Obrázek 4.13: Binarizovaný průběh jasových intenzit s nalezenou oblastí čárového kódu.

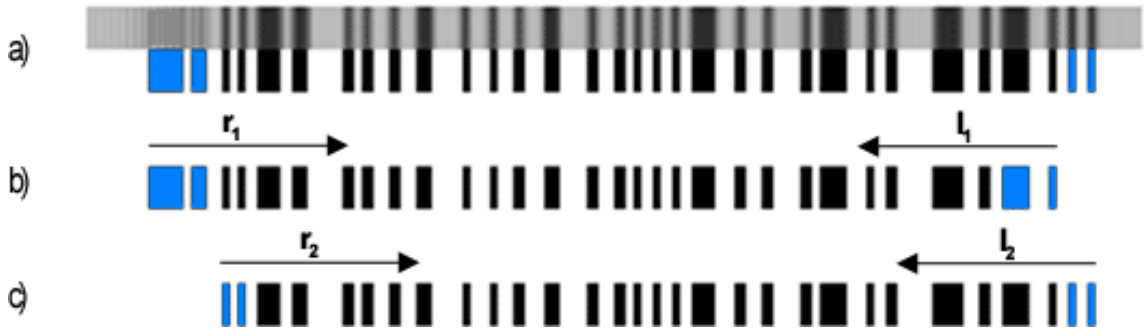
4.3.2 Čtení bloků šířek

Čárový kód EAN-13 je tvořen celkově z 59 čar a mezer (30 čar a 29 mezer). Vlivem šumu v obraze se může stát, že hranice budou chybně vymezeny (viz obr. 4.14 (a)) a čárový kód pak bude obsahovat více, či méně požadovaných čar a mezer. V tomto případě budou ztraceny některé důležité informace a dekódovací proces nemůže být proveden. V opačném případě je nutné vhodně vybrat, či rekonstruovat čáry a mezery, které pak projdou procesem dekódování.

Z hlediska nutné nízké výpočetní náročnosti pro zpracování v reálném čase nejsou prováděny žádné složité operace, ale je použito k dekódování pouze prvních a také posledních 59 čar a mezer (viz obr. 4.14 (bc)). To vychází z předpokladu, že alespoň jedna hranice ať už pravá, či levá je vymezena správně.

Při čtení je nutné zohlednit fakt, že čárový kód může být v obraze převrácen a proto je třeba číst binarizované čáry a mezery nejenom zleva doprava, ale také i v opačném směru.

Celkově je tedy nutné dekódovat dvě posloupnosti šířek čar a mezer (viz obr. 4.14 (bc)) ve dvou směrech a poté vybrat validní dekódovanou informaci. Z tabulky 2.1 ze sloupce **vzor** však můžeme vypozorovat, že první kódovaná číslice vždy náleží kódovací sadě **L**. Pokud se tak nestane, pak není nutné provádět další dekódování v daném směru, neboť by to stejně směřovalo na nevalidní dekódovanou informaci.



Obrázek 4.14: (a) Špatně vytyčené hranice binarizovaného kódu; (b) Dekódování posloupnosti prvních 59 čar a mezer v obou směrech; (c) Dekódování posloupnosti posledních 59 čar a mezer v obou směrech.

4.3.3 Metoda dekódování

Metod a možností dekódování se nabízelo velké množství. Nešlo však přesně odhadnout, která bude nejlepší, a proto je bylo nutné vyzkoušet. Mnou zkoušené metody je možné rozdělit do dvou kategorií:

- Dekódování neuronovou sítí
- Dekódování nalezením nejlepší shody

Dekódování neuronovou sítí

Při dekódování neuronovou sítí byla použita dopředná neuronová síť s jednou skrytou vrstvou, spojitou sigmoidální aktivační funkcí a metodou učení Backpropagation.

Byly použity dvě základní konfigurace neuronové sítě, které se lišily pouze vstupy. U první verze jsem se inspiroval článkem [14] a použil jsem 56-ti bitový vstup. Vstupní vektor pak používal obdobné kódování jako v tabulce 4.1, kde délky posloupností jedniček a nul byly zvoleny úměrně k reálným šířkám čar a mezer. V druhé verzi měla neuronová síť osm vstupů. Vstupní vektor byl pak tvořen posloupností dvojic, kde první hodnota určovala, zda se jedná o mezeru (hodnota 0), či čáru (hodnota 1), za níž následovala normalizovaná šířka čáry, či mezery. Například pro kód 1110010 je vstupní vektor $[1, \frac{3}{7}, 0, \frac{2}{7}, 1, \frac{1}{7}, 0, \frac{1}{7}]$.

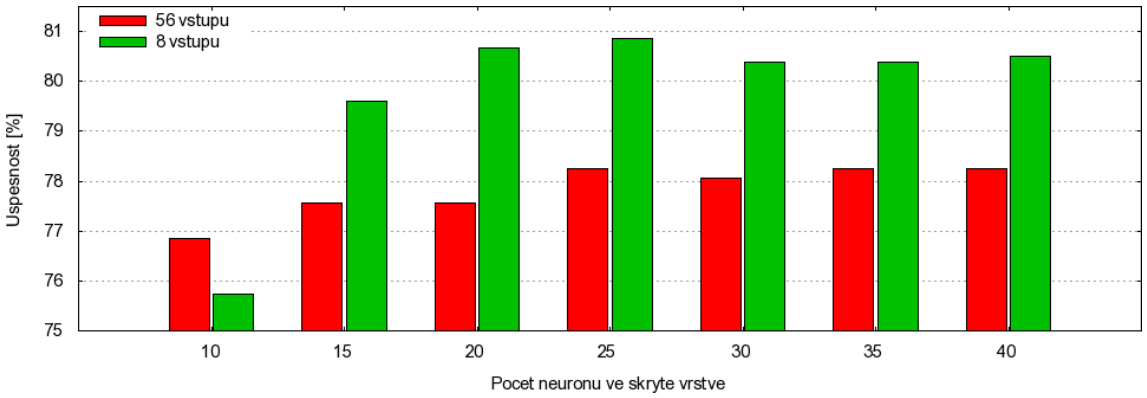
Obě konfigurace měly sedm binárních výstupů. Výstupní vektor tedy představoval binárně zakódovanou číslici dle tabulky 4.1.

Neuronové sítě byly trénovány na 10000 vygenerovaných vzorcích dat s náhodnou změnou šířek čar a mezer od 0% do 12,5% vůči ideálním kódovaným šířkám číslic. Právě hodnota změny 12,5% je maximální možnou hodnotou, při které je ještě možné správně dekódovat.

Pro testovací účely neuronových sítí byla použita reálná data. Jelikož získání vhodných reálných dat se jevilo jako velmi pracné, byla data získána automaticky samotnou aplikací. Celkem bylo získáno 1764 vzorků reálných dat. Takové automatické získání vzorků dat má jistou nevýhodu. Tou je získání i chybných vzorků, tedy takových dat, kde vstupům neodpovídají požadované výstupy, což je zapříčiněno špatnou binarizací a následně lokalizací hranic čárového kódu na nekvalitních snímcích.

Testování bylo prováděno na neuronových sítích s různým počtem neuronů ve skryté vrstvě. Výsledky jsou graficky znázorněny na obrázku 4.15. Při konfiguraci s 10 neurony ve skryté vrstvě měla vyšší úspěšnost neuronová síť s 56-ti vstupy, neboť obsahovala více

neuronů než 8-vstupová neuronová síť a tím se dokázala lépe naučit. Ovšem pro skryté vrstvy od 15-ti do 40-ti neuronů je výrazně úspěšnější 8-vstupová neuronová síť. Je to zřejmě způsobeno tím, že u 56-vstupové neuronové sítě je nutné šířky čar a mezer vhodně „napasovat“ do 56-ti bitové posloupnosti, což v některých případech vede k méně přesné informaci o šířkách čar a mezer.



Obrázek 4.15: Úspěšnost neuronových sítí v různých konfiguracích.

Dekódování nalezením nejlepší shody

Tato metoda spočívá v porovnávání přečteného bloku (dvou čar a dvou mezer jež kódují číslici) se všemi referenčními kódy. Pro každý referenční kód r je vypočtena míra diference $f(r,s)$, která je dána vztahem 4.7, kde w_{ri} je i-tá šířka čáry resp. mezery referenční kódované číslice a w_{si} je i-tá šířka čáry resp. mezery v přečteném bloku. Odpovídající dekódovaná číslice r je pak určena nalezením minimální $f(r,s)$.

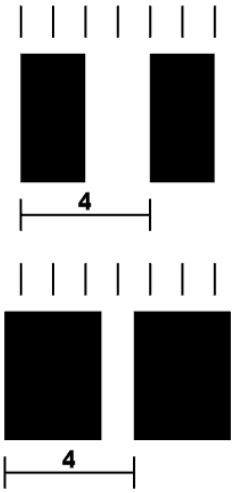
$$f(r,s) = \sum_{i=1}^{4} |w_{ri} - w_{si}| \tag{4.7}$$

Je nutno podotknout, že jednotlivé šířky čar, či mezer nemusí být chápány pouze jako kladné hodnoty. Další možností je k jednotlivým šířkám také přidat informaci o tom, zda se jedná o čáru nebo mezeru. To může být realizováno kladnou hodnotou šířky u čar a zápornou hodnotou šířky u mezer (viz tab. 4.1).

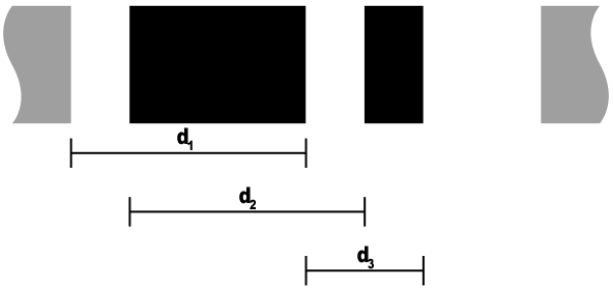
V [22] je zmíněno, že hlavním problémem zkreslení, kterými se potýkají laserové čtečky je způsoben rozpíjením inkoustu při tisku čárového kódu. U tohoto zkreslení se zdají být širší čáry než-li mezery. Elegantním řešením tohoto problému je to, že neprobíhá dekódování porovnáváním jednotlivých šířek čar a mezer, ale místo toho porovnávání vzdáleností mezi podobnými hranami[2]. Celou záležitost objasňuje obrázek 4.16 na němž jsou dvě binarizované části čárového kódu, u kterých je měřena vzdálenost mezi podobnými hranami. Nahoře jsou binarizované čáry u kterých nedošlo k rozpíjení. V tomto případě je vzdálenost mezi podobnými hranami rovna čtyřem jednotkám. Stejně je tomu i na obrázku dole, kde došlo k rozpíjení inkoustu. Z toho plyne, že při rozšiřování čar jsou stejnou mírou zužovány sousední mezery, což v konečném důsledku dává vždy stejnou vzdálenost mezi podobnými hranami.

[2]Vzdálenost mezi podobnými hranami je též nazývána jako Δ-vzdálenost (Δ-*distance*), či t-vzdálenost (*t-distance*). Je to v podstatě suma šířky čáry a přilehlé mezery.

Způsob dekódování jedné číslice je znázorněn na obrázku 4.17. Tomu také odpovídají referenční kódy v tabulce 4.1 ve sloupci Δ-vzdálenosti. V tabulce si lze všimnout, že nastávají kolize mezi kódovanými dvojicemi číslic $1-7$ a $2-8$. Tyto problémové kódy je pak nutné dekódovat postupným porovnáním jednotlivých šířek.



Obrázek 4.16: Řešení problému s rozpíjením inkoustu.



Obrázek 4.17: Způsob dekódování dle Δ-vzdáleností.

Výsledky úspěšnosti všech metod dekódování jsou zaznamenány v grafu na obrázku 4.18. Všechny metody byly testovány na stejné testovací množině o které jsem se zmínil u neuronových sítí v sekci 4.3.3. Z grafu je patrné, že nejlepších výsledků dosahuje dekódování nalezením nejlepší shody.

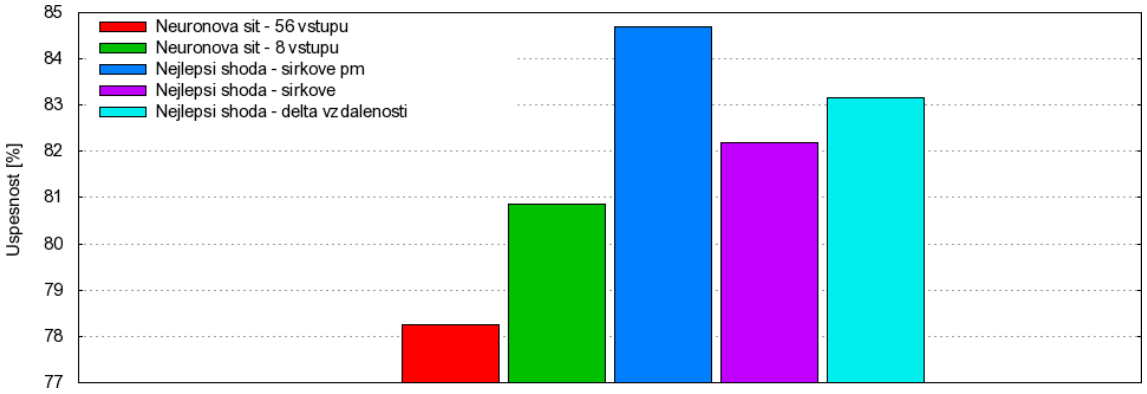
V případě šířkového kódování s kladnými a zápornými hodnotami šířek je dosaženo vyšší úspěšnosti než u kódování pouze s kladnými šířkami. To je způsobeno tím, že přidaná informace o čárách a mezerách v podobě kladných a záporných šířek způsobuje vyšší diference mezi porovnávanými referenčními kódy. V druhém pořadí úspěšnosti je kódování pomocí Δ-vzdáleností. Zde hraje velkou roli vzájemná závislost čar a mezer. V případě čárového kódu zaznamenaného v nízkém rozlišení dochází k méně přesné binarizaci a vzájemná závislost čar a mezer nejspíše způsobuje vyšší chybovost než u dekódování dle jednotlivých šířek čar a mezer. Dekódování pomocí Δ-vzdáleností má zřejmě své opodstatnění u laserových čteček čárových kódů, kde je dosahováno vyšší přesnosti resp. vyššího rozlišení.

Tabulka 4.1: Kódovací sady čárového kódu EAN-13 - různé možnosti chápání kódů.

Číslice	**Kódovací sada L**			
	binárně	šířkově	šířkově ±	Δ-vzdálenosti
0	0001101	3,2,1,1	-3,+2,-1,+1	5,3,2
1	0011001	2,2,2,1	-2,+2,-2,+1	4,4,3
2	0010011	2,1,2,2	-2,+1,-2,+2	3,3,4
3	0111101	1,4,1,1	-1,+4,-1,+1	5,5,2
4	0100011	1,1,3,2	-1,+1,-3,+2	2,4,5
5	0110001	1,2,3,1	-1,+2,-3,+1	3,5,4
6	0101111	1,1,1,4	-1,+1,-1,+4	2,2,5
7	0111011	1,3,1,2	-1,+3,-1,+2	4,4,3
8	0110111	1,2,1,3	-1,+2,-1,+3	3,3,4
9	0001011	3,1,1,2	-3,+1,-1,+2	4,2,3

Číslice	**Kódovací sada G**			
	binárně	šířkově	šířkově ±	Δ-vzdálenosti
0	0100111	1,1,2,3	-1,+1,-2,+3	2,3,5
1	0110011	1,2,2,2	-1,+2,-2,+2	3,4,4
2	0011011	2,2,1,2	-2,+2,-1,+2	4,3,3
3	0100001	1,1,4,1	-1,+1,-4,+1	2,5,5
4	0011101	2,3,1,1	-2,+3,-1,+1	5,4,2
5	0111001	1,3,2,1	-1,+3,-2,+1	4,5,3
6	0000101	4,1,1,1	-4,+1,-1,+1	5,2,2
7	0010001	2,1,3,1	-2,+1,-3,+1	3,4,4
8	0001001	3,1,2,1	-3,+1,-2,+1	4,3,3
9	0010111	2,1,1,3	-2,+1,-1,+3	3,2,4

Číslice	**Kódovací sada R**			
	binárně	šířkově	šířkově ±	Δ-vzdálenosti
0	1110010	3,2,1,1	+3,-2,+1,-1	5,3,2
1	1100110	2,2,2,1	+2,-2,+2,-1	4,4,3
2	1101100	2,1,2,2	+2,-1,+2,-2	3,3,4
3	1000010	1,4,1,1	+1,-4,+1,-1	5,5,2
4	1011100	1,1,3,2	+1,-1,+3,-2	2,4,5
5	1001110	1,2,3,1	+1,-2,+3,-1	3,5,4
6	1010000	1,1,1,4	+1,-1,+1,-4	2,2,5
7	1000100	1,3,1,2	+1,-3,+1,-2	4,4,3
8	1001000	1,2,1,3	+1,-2,+1,-3	3,3,4
9	1110100	3,1,1,2	+3,-1,+1,-2	4,2,3



Obrázek 4.18: Shrnutí úspěšnosti zmiňovaných metod dekódování.

Kapitola 5

Implementace

Tato kapitola je členěna na dvě části. V první části jsou rozebrány implementační prostředky pomocí nichž bylo dosáhnuto výsledného systému pro dekódování čárového kódu. Druhá část je věnována stručnému popisu implementovaného systému z pohledu tříd.

5.1 Implementační prostředky

Jak již bylo v úvodu zmíněno, aplikace by měla v reálném čase dekódovat snímky s čárovým kódem. Nutností je tedy vysoká efektivita aplikace při zpracovávání jednotlivých snímků. Z tohoto důvodu byl vybrán programovací jazyk *C++*, který má prvky nízké úrovně, jež umožňují rychlou práci s daty a další výhodou je, že také obsahuje prvky objektově orientovaného programování.

Ideální volbou k programovacímu jazyku *C++* bylo použití knihovny *OpenCV*, která obsahuje podpůrné prvky pro zpracování obrazu ať už statického, či dynamického. Má velmi širokou podporu obrazových formátů a poskytuje také velmi jednoduché grafické uživatelské rozhraní.

Celá aplikace byla napsána ve vývojovém prostředí *Microsoft Visual Studio 2005*, neboť poskytuje velmi kvalitní editor kódu s integrovaným debuggerem.

5.1.1 Programovací jazyk C++

C++ je staticky typový, kompilovaný, objektově orientovaný programovací jazyk. Má podporu několika programovacích stylů (paradigmat) jako je procedurální programování, objektově orientované programování a generické programování. Není tedy jazykem čistě objektovým.

Byl vytvořen jako vylepšení jazyka *C*. Kombinuje tedy nízkoúrovňové programování jazyka C s prvky objektově orientovaného programování.

Jeho standardní knihovna je složena ze dvou částí. První část obsahuje jádro jazyka, jež je mírně modifikovanou verzí standardní knihovny jazyka *C*. Druhou část tvoří *Standard Template Library* (STL), která obsahuje velké množství užitečných datových struktur a algoritmů. Jsou to například vektory (vylepšené pole), spojové seznamy, iterátory, zobecněné ukazatele, (multi)mapy a (multi)sety.

C++ je velmi populární a vysoce efektivní programovací jazyk.

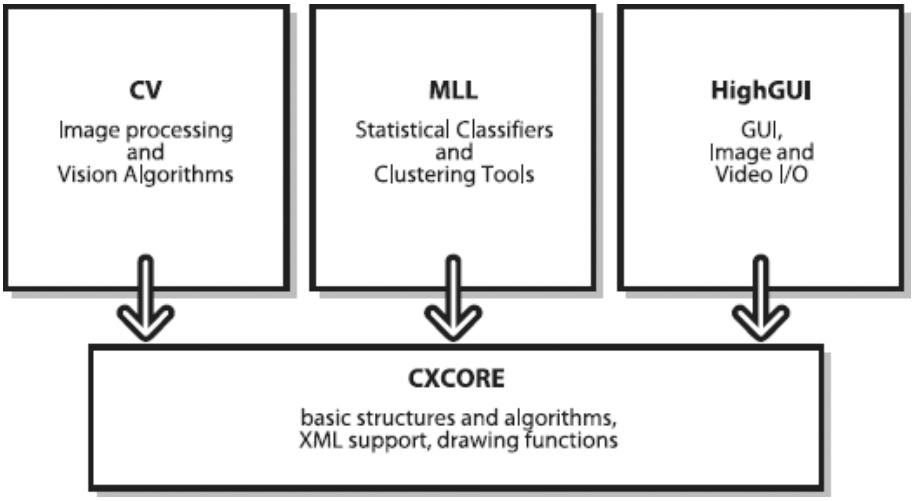
5.1.2 Knihovna OpenCV

OpenCV (Open Source Computer Vision) je volně dostupná knihovna, která je zaměřena především na počítačové vidění a zpracování obrazu v reálném čase. Knihovna je multiplatformní a existuje pod operačními systémy *Windows, Linux* i *Mac OS X*. Převážně je dostupná pro programovací jazyky *C/C++*, ale má i aktivní vývoj na rozhraní pro *Python, Ruby, Matlab* a jiné programovací jazyky.

Knihovna je dělena do několika základních modulů, které mají specifickou funkčnost. Jedná se o moduly [6]:

- ***CXCORE*** - Obsahuje datové struktury a funkce pro maticovou algebru, datovou transformaci, uchovávání dat, zpracování chyb, správu paměti a kreslení základních tvarů.
- ***CV*** - Obsahuje funkce pro zpracování obrazu, analýzu obrazu, rozpoznání vzorů obrazu a funkce pro kalibraci kamery.
- ***Machine Learning (ML)*** - Obsahuje funkce pro strojové učení zahrnující shlukování, klasifikaci a analýzu dat.
- ***HighGUI*** - Uživatelské rozhraní, ukládání a načítání obrazových dat a videa.
- ***CVCAM*** - Funkce pro práci s kamerou.
- ***CvAux*** - Obsahuje experimentální, ale i zavržené funkce.

Struktura knihovny je ilustrována na obrázku 5.1. Do této struktury však není zahrnut modul *CvAux*.



Obrázek 5.1: Struktura knihovny OpenCV [6].

5.1.3 Microsoft Visual Studio

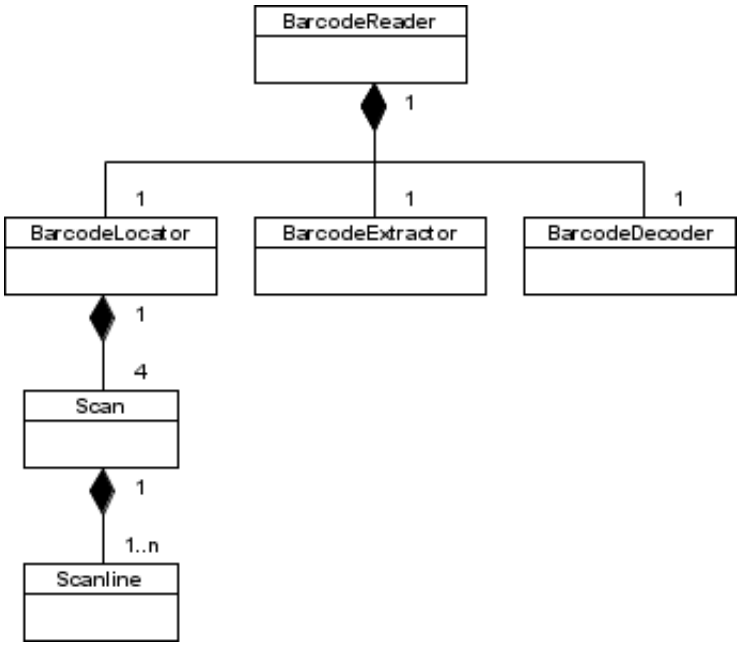
Microsoft Visual Studio (MVS) je vývojové prostředí od firmy *Microsoft*. Může být použito k vývoji konzolových aplikací, ale také i aplikací s grafickým uživatelským rozhraním

pro platformy *Microsoft Windows, Windows Mobile, Windows CE, .NET* a *Microsoft Silverlight*. MVS má podporu programovacích jazyků prostřednictvím jazykových služeb, což umožňuje, aby editor kódu a debugger podporoval jakýkoliv programovací jazyk. Mezi vestavěné jazyky však patří *C/C++, VB.NET* a *C#*.

Toto vývojové prostředí je vybaveno pokročilými vývojovými nástroji, které umožňují pohodlný a rychlý vývoj aplikací.

5.2 Implementační detaily

Celý systém pro dekódování čárového kódu je tvořen objekty pěti tříd, jejichž hierarchie je znázorněna na obrázku 5.2.



Obrázek 5.2: Hierarchie tříd.

5.2.1 Třída BarcodeReader

Je hlavní třídou celého systému, která „zastřešuje“ všechny ostatní třídy. Klíčovou metodou této třídy je metoda `getCode()`, která zpracuje obraz s čárovým kódem a jejím výstupem je pak dekódovaná informace v podobě řetězce číslic.

Vyšší úspěšnost dekódování je zajištěna dekódováním většího počtu přečtených průběhů jasových intenzit obrazu vedoucích skrz nalezenou polohu čárového kódu (viz obr. 5.3). Tímto však může být získána skupina i více různých validních dekódovaných řetězců číslic. Problém výběru dekódovaného řetězce je vyřešen tak, že s každou číslicí je vázána hodnota podobnosti vůči referenční kódované číslici. Celková suma podobností číslic je pak rozhodujícím faktorem při výběru validního dekódovaného řetězce.

Celý algoritmus metody `getCode()` může být stručně shrnut do následujících kroků:

1. Nalezni polohu P a orientaci 0 čárového kódu v obrazu I.
2. Dokud počet dekódování $D > 0$, opakuj:

(a) Získej posloupnost S šířek čar a mezer čárového kódu z obrazu I dle polohy P a orientace O.

(b) Dekóduj posloupnost S šířek a případný validní kód K ulož do pole validních kódů PK.

(c) Změň orientaci 0 o 1°.

(d) $D = D$ - 1

3. Vyber nejvhodnější validní kód K z pole validních kódů PK.

Obrázek 5.3: Zvýšení úspěšnosti dekódování; Vlevo: Neúspěšné dekódování z jednoho přečteného průběhu jasové intenzity; Vpravo: Úspěšné dekódování zapříčiněné dekódováním více průběhů jasových intenzit.

5.2.2 Třída BarcodeLocator

Provádí lokalizaci čárového kódu v obrazu. Obsahuje čtyři objekty třídy `Scan` pro hledání podobných význačných bodů v orientacích 0°, 45°, 90° a 135°.

Celý lokalizační proces realizuje metoda `localize()`, která provede aktualizaci všech objektů třídy `Scan` pro konkrétní vstupní obraz a poté získá polohu a orientaci čárového kódu z jednoho objektu, který dosahoval nejvyššího skóre nalezených podobností.

5.2.3 Třída Scan

Provádí hledání podobných význačných bodů mezi sousedními řádky načtených průběhů intenzit jasu (třída `Scanline`). Hlavní metodou je metoda `doScan()`, která zpracovává data v následujících krocích:

1. Proveď aktualizaci všech objektů třídy `Scanline` pro konkrétní obraz.
2. Nalezni podobné význačné body v datech zpracovaných objekty třídy `Scanline`.
3. Vypočítej skóre a posun nalezených význačných bodů.

Metoda `getLocation()` pak realizuje nalezení nejdelší posloupnosti podobných význačných bodů ze které je následně vypočtena poloha čárového kódu.

5.2.4 Třída Scanline

Je na nejnižší úrovni v hierarchii lokalizace. Hlavní metodou je metoda `update()` ve které dojde k přečtení posloupnosti intenzit jasu obrazu (metoda `getIntensities()`) a poté následnému vyhledání význačných bodů (metoda `findFeatures()`).

5.2.5 Třída BarcodeExtractor

Je „mostem“ mezi lokalizací a dekódováním. Umožňuje přečíst posloupnost jasových intenzit obrazu (metoda `readLine()`) a následně tuto posloupnost binarizovat dynamickým prahováním se subpixelovou přesností (metoda `setBars()`). Poté je binarizovaná posloupnost „vyčištěna“ od zbytečných čar a mezer, které nenáleží čárovému kódu (metoda `cleanBars()`). Výstupem je posloupnost šířek čar a mezer určená k dekódování.

5.2.6 Třída BarcodeDecoder

Je posledním článkem systému. Hlavní metodou je metoda (`decode()`), která má dva vstupní parametry. Prvním je pole šířek čar a mezer, které pochází z objektu třídy `BarcodeExtractor` a druhým parametrem je směr dekódování. Výsledkem je pole dekódovaných číslic, kde s každou číslicí je svázána míra podobnosti vůči její referenční zakódované číslici. V posledním kroku už jen dojde k ověření valididy dekódovaných číslic dle kontrólního součtu.

Kapitola 6

Dosažené výsledky

Tato kapitola se zabývá testováním vytvořené aplikace, analýzou případných chyb a obsahuje též popis testovací sady. Dovršením kapitoly je porovnání výsledků s komerčními aplikacemi zaměřenými na dekódování čárového kódu.

6.1 Testovací sada

Testovací snímky byly pořízeny fotoaparátem Canon PowerShot A40. Celkem bylo nafoceno 500 snímků s čárovými kódy typu EAN-13, které byly uloženy do formátu JPG[1] v 8-bitové barevné hloubce o rozměrech 640×480 pixelů.

Snímky pocházely z 50-ti produktů a na každý produkt tak připadalo 10 snímků. Čárové kódy na snímcích zaujímaly cca 2-15% z celkové plochy obrazu. Byly nafoceny v různých orientacích i z perspektivy a v různé kvalitě (rozmazané, ostré, s odlesky, s deformacemi, podexponované atd.). Některé snímky dokonce obsahovaly i více čárových kódů. Pro názornost je nepatrný zlomek nafocených snímků uveden na obrázku 6.2.

Každý snímek byl řádně anotován. Snímku s názvem `aaaaaaaaaaaaa-bb.jpg` náleží anotační soubor `aaaaaaaaaaaaa-bb.jpg.txt`, kde `aaaaaaaaaaaaa` představuje třináctimístný kód čárového kódu a `bb` je pořadové číslo snímku daného produktu. Ukázka obsahu anotačního souboru je uvedena na obrázku 6.1. Anotační soubor obsahuje příslušný třináctimístný kód a také souřadnice čtyř bodů, jež ohraničují čárový kód. Tyto souřadnice pak slouží pro testování lokalizačního algoritmu.

```
code:8713439153521
   x:281 y:149
   x:463 y:238
   x:407 y:337
   x:236 y:238
```

Obrázek 6.1: Ukázka obsahu anotačního souboru.

[1] Standardní metoda ztrátové komprese používané pro ukládání počítačových obrázků ve fotorealistické kvalitě.

Další součást testovací sady tvořily videosekvence, které byly v rozlišení 640×480 pixelů s frekvencí snímků 10fps. Celkově byly pořízeny tři videosekvence s předměty resp. produkty obsahující čárový kód:

- `posun.avi` - Pohyb předmětu (109 snímků ve videosekvenci).
- `zoom.avi` - Přiblížení a oddálení předmětu (101 snímků ve videosekvenci).
- `rotace.avi` - Otáčení předmětem (103 snímků ve videosekvenci).

Obrázek 6.2: Ukázka některých snímků z testovací sady.

6.2 Test lokalizace

Lokalizační algoritmus byl otestován na 500 snímcích testovací sady o kterých již bylo zmíněno v podkapitole 6.1. Celkem byly provedeny čtyři testy, které se lišily pouze ve změně parametru rozteče mezi skenovanými řádky.

Výsledky testů jsou uvedeny v tabulce 6.1. Z tabulky je patrná jistá závislost rozteče skenovaných řádku na úspěšnosti lokalizace. Nejvyšší úspěšnosti $(94,8\%)$ bylo dosaženo při rozteči 5 pix mezi skenovanými řádky a naopak nejhorší úspěšnosti $(84,4\%)$ při rozteči 20 pix.

Na obrázku 6.3 je znázorněna správná lokalizace čárového kódu. Nutnou podmínkou správné lokalizace je, aby byl lokalizační bod (modrý křížek) situován mezi levou a pravou hranicí (zelená čára) čárového kódu. Další podmínka byla kladena na určení správné orientace čárového kódu - červená čára znázorňující orientaci musela protínat levou a pravou hranici čárového kódu.

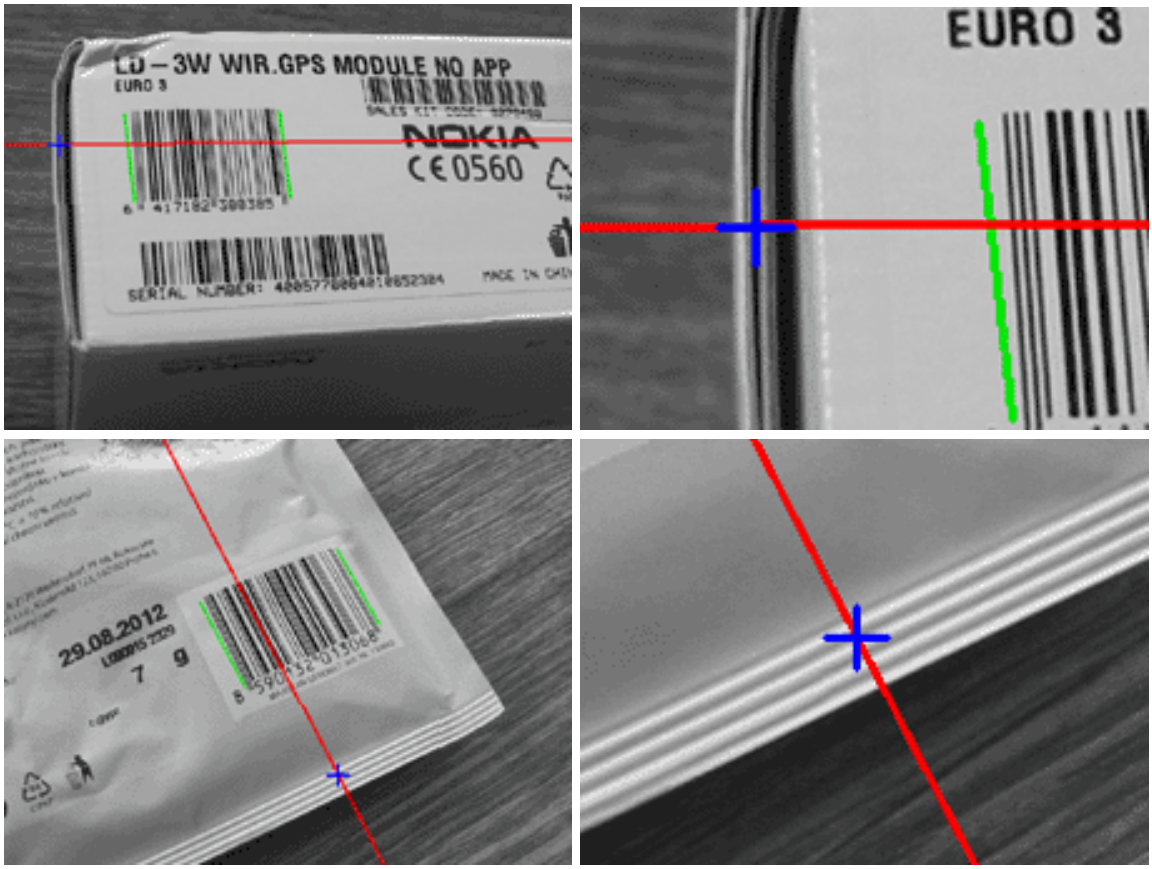
Chybná lokalizace je ilustrována obrázkem 6.4. K té dochází pokud se v obrazu vyskytuje souvislá oblast pomyslných čar a mezer, které jsou vzájemně rovnoběžné. Je-li délka těchto pomyslných čar a mezer větší než výška čárového kódu, pak dochází k chybné lokalizaci.

Méně častý problém nastává u čárových kódů s nízkou výškou (viz obr. 6.5), kde je splněna podmínka polohy lokalizačního bodu, ale orientace je špatně vypočítána v důsledku malé výšky čárového kódu.

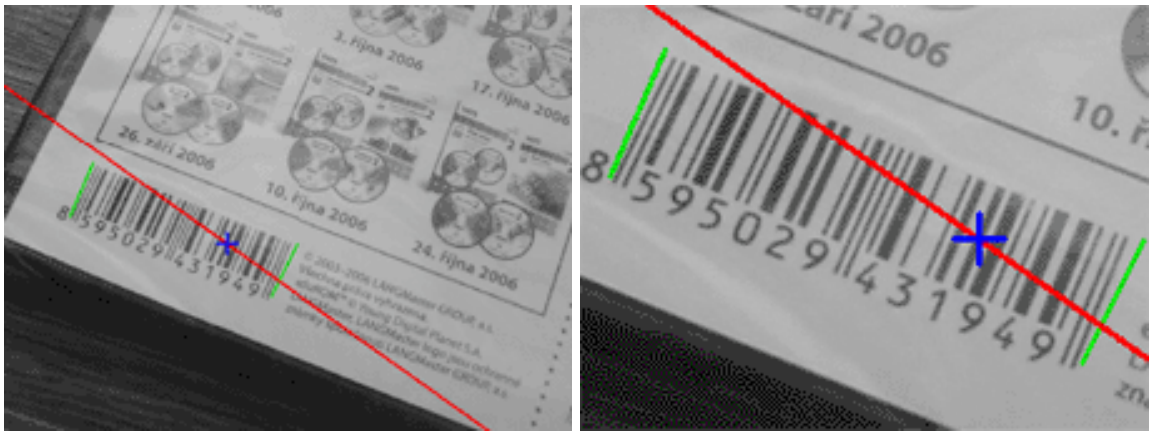
Tabulka 6.1: Úspěšnost lokalizace.

Rozteč řádků [pix]	**Správně lokalizovaných**	**Úspěšnost [%]**
5	474/500	94,8
10	459/500	91,8
15	446/500	89,2
20	420/500	84,0

Obrázek 6.3: Správná lokalizace.



Obrázek 6.4: Chybná lokalizace.



Obrázek 6.5: Částečná lokalizace.

6.3 Test dekódování

Z hlediska objektivního posouzení úspěšnosti dekódování byly do testování zahrnuty pouze snímky, u kterých došlo ke správné lokalizaci. Jednalo se tedy o 474 snímků, které byly úspěšně lokalizovány skenovanými řádky s roztečí 5 pix. Dekódování probíhalo pouze na jednom přečteném průběhu jasových intenzit. Úspěšně bylo dekódováno 264 snímků, což je

v přepočtu 55,7% úspěšnost. U zbývajících 210 problematických snímků byly zkoumány příčiny chybného dekódování. Tyto příčiny byly rozděleny do tří skupin:

1. Dekódování bylo nekorektní v důsledku zkreslení šířek čar a mezer, což se většinou projevovalo u velmi drobných čárových kódů, jejichž šířky čar byly na hranici jednoho pixelu. Tento problém ilustruje obrázek 6.6, na kterém je znázorněn průběh jasové intenzity v podobě pruhu a křivky s nalezenými lokálními extrémy. Pod pruhem jasových intenzit se nachází binarizované čáry, které náleží čárovému kódu. U některých binarizovaných čar si lze všimnout, že nemají odpovídající šířku, což způsobuje chybu při dekódování.

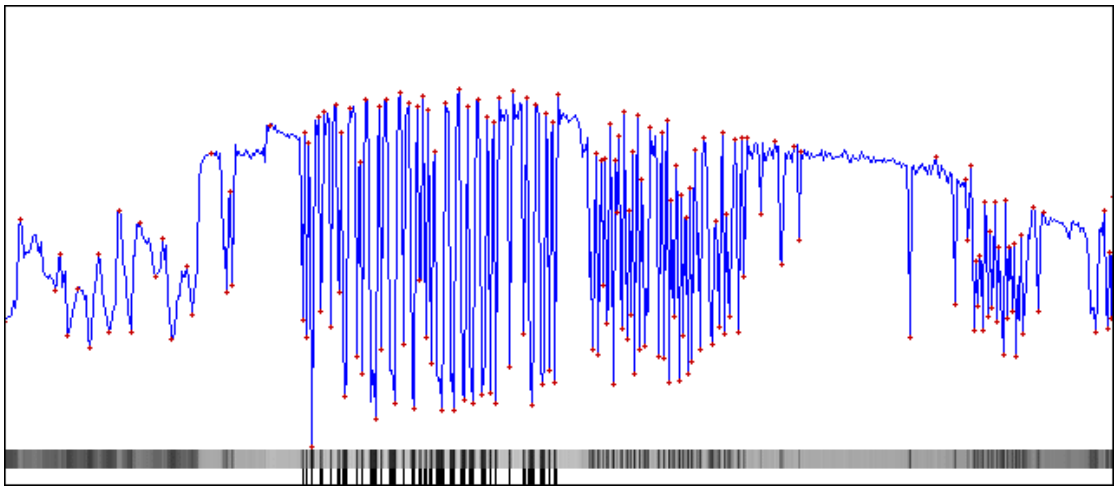
2. Dekódování bylo neúspěšné v důsledku chybějících, či přebývajících čar. K nadbytečnému množství čar docházelo vlivem šumu v obrazu a naopak k malému množství docházelo splynutím čar u podexponovaných snímků, nebo ztrátou u přeexponovaných snímků (odlesky). Tento problém prezentuje obrázek 6.7, kde došlo ke správnému vymezení hranic čárového kódu, ovšem uvnitř čárového kódu došlo ke ztrátě některých čar vlivem odlesku.

3. Dekódování bylo neúspěšné v důsledku špatného vytyčení hranic čárového kódu, což se většinou projevovalo u podexponovaných snímků, kdy docházelo k splynutí některých čar. Tento problém je ilustrován na obrázku 6.8.

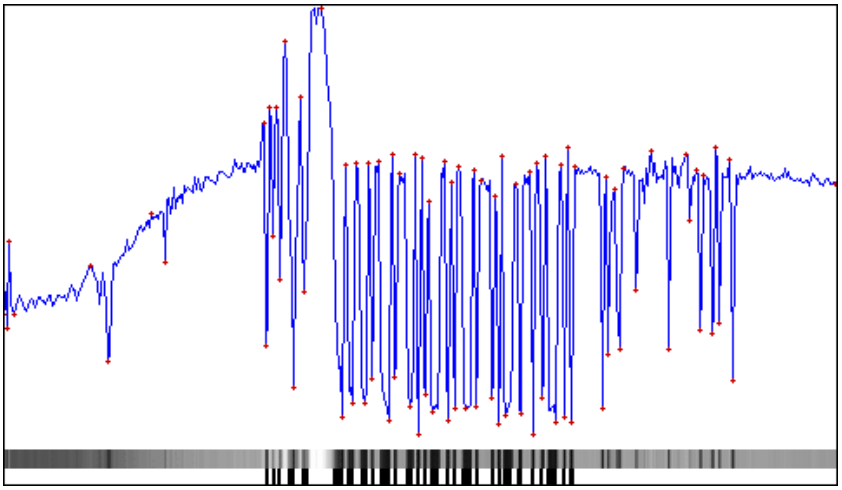
Výsledky chybovosti zde zmíněných příčin jsou uvedeny v tabulce 6.2.

Tabulka 6.2: Chybovost při dekódování.

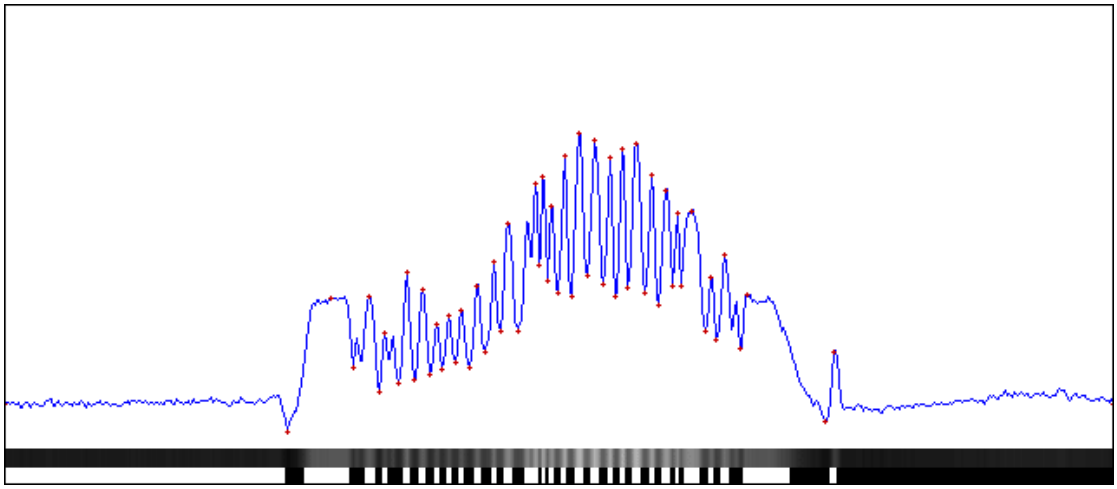
Skupina	**Chybně dekódovaných**	**Chybovost [%]**
1.	113/474	23,8
2.	45/474	9,5
3.	52/474	10,8



Obrázek 6.6: Neodpovídající šířky čar způsobují chybu dekódování.



Obrázek 6.7: Chyba dekódování v důsledku ztráty čar a mezer.



Obrázek 6.8: Chyba dekódování v důsledku špatného určení hranic čárového kódu.

6.4 Test celého systému

Jak již název podkapitoly napovídá, byla testována lokalizace a dekódování jako jeden celek. U všech provedených testů byla vyhodnocena úspěšnost, chybovost a časová náročnost na zpracování jednoho snímku.

Pro úplnost je úspěšností rozuměno, že dekódovaná informace odpovídala čárovému kódu. Naopak chybovost je definována, jako úspěšně provedené dekódování (validní dekódovaná informace dle kontrolního součtu), ale dekódovaná informace neodpovídá čárovému kódu. Časová náročnost byla měřena jako interval mezi spuštěním a ukončením aplikace. K časové náročnosti je také nutno podotknout, že testy byly prováděny na notebooku s 2GHz procesorem AMD Turion 64 X2.

Testování systému probíhalo nejdříve na 500 snímcích testovací sady. Jednotlivé testy se pak lišily ve dvou důležitých parametrech, dle kterých lze ovlivnit úspěšnost a časovou náročnost. Jednalo se o rozteč mezi skenovanými řádky pro lokalizační algoritmus a počet přečtených řádků obrazu z nichž se provádí dekódování. Výsledky těchto testů jsou shrnuty

v tabulkách 6.3 až 6.6, kde nejvyšší úspěšnosti 74,6% bylo dosaženo při 5-pixelové lokalizační rozteči a dvaceti dekódovaných řádcích. Ovšem tato úspěšnost si vybírá svou daň v podobě vyšší časové náročnosti a vyšší chybovosti. Průměrné hodnoty pak činily pro úspěšnost 66,6%, chybovost 6,4% a čas zpracování 152ms.

Další testy byly prováděny na videosekvencích z testovací sady a byly především zaměřeny na zhodnocení časové náročnosti. Testování probíhalo s pevně zvoleným parametrem 10 dekódovaných řádků a se změnou v lokalizačních roztečích 5, 10, 15 a 20 pixelů. Výsledky testovaných videosekvencí jsou uvedeny v tabulkách 6.7 až 6.10.

Průměrná časová náročnost ve videosekvencích dosahovala v porovnání s dekódováním snímků vždy lepších výsledků, což se také od navrženého systému očekáváno. Co se týče úspěšnosti a chybovosti, tak uspokojivých výsledků dosahovalo video `rotace.avi` a naopak nedostačujících výsledků `posun.avi`.

Tabulka 6.3: Výsledky dekódování snímků při lokalizační rozteči 5 pix.

Dek. řádků	**Úspěšnost** [%]	**Chybovost** [%]	Čas [ms]
1	54,8	1,6	242
5	70,4	4,2	253
10	74,6	5,6	261
15	74,4	7,2	275
20	74,6	9,4	281

Tabulka 6.4: Výsledky dekódování snínků při lokalizační rozteči 10 pix.

Dek. řádků	**Úspěšnost** [%]	**Chybovost** [%]	Čas [ms]
1	53,4	3,6	133
5	68,0	6,6	138
10	72,2	6,8	145
15	72,6	9,2	154
20	72,6	10,4	161

Tabulka 6.5: Výsledky dekódování snímků při lokalizační rozteči 15 pix.

Dek. řádků	**Úspěšnost** [%]	**Chybovost** [%]	Čas [ms]
1	51,4	1,8	93
5	65,2	4,0	100
10	69,2	6,4	107
15	69,8	8,4	122
20	71,2	10,2	126

Tabulka 6.6: Výsledky dekódování snímků při lokalizační rozteči 20 pix.

Dek. řádků	**Úspěšnost** [%]	**Chybovost** [%]	Čas [ms]
1	50,2	1,6	76
5	63,2	4,8	82
10	67,2	6,8	89
15	67,8	9,0	96
20	68,6	10,8	103

Tabulka 6.7: Výsledky dekódování videosekvencí při lokalizační rozteči 5 pix.

Video	**Úspěšnost** [%]	**Chybovost** [%]	Čas [ms/sn.]
posun.avi	9,2	28,4	224
zoom.avi	25,7	6,9	229
rotace.avi	60,2	20,4	239
Průměr	31,3	18,8	231

Tabulka 6.8: Výsledky dekódování videosekvencí při lokalizační rozteči 10 pix.

Video	**Úspěšnost** [%]	**Chybovost** [%]	Čas [ms/sn.]
posun.avi	11,0	28,4	117
zoom.avi	26,7	15,8	134
rotace.avi	72,8	12,6	143
Průměr	36,4	19,2	131

Tabulka 6.9: Výsledky dekódování videosekvencí při lokalizační rozteči 15 pix.

Video	**Úspěšnost** [%]	**Chybovost** [%]	Čas [ms/sn.]
posun.avi	5,5	37,6	108
zoom.avi	23,8	20,8	112
rotace.avi	63,1	26,2	98
Průměr	30,4	28,4	106

Tabulka 6.10: Výsledky dekódování videosekvencí při lokalizační rozteči 20 pix.

Video	**Úspěšnost** [%]	**Chybovost** [%]	Čas [ms/sn.]
posun.avi	18,3	15,6	73
zoom.avi	28,7	12,9	79
rotace.avi	67,0	24,3	85
Průměr	37,7	17,6	79

6.5 Srovnání s komerčními aplikacemi

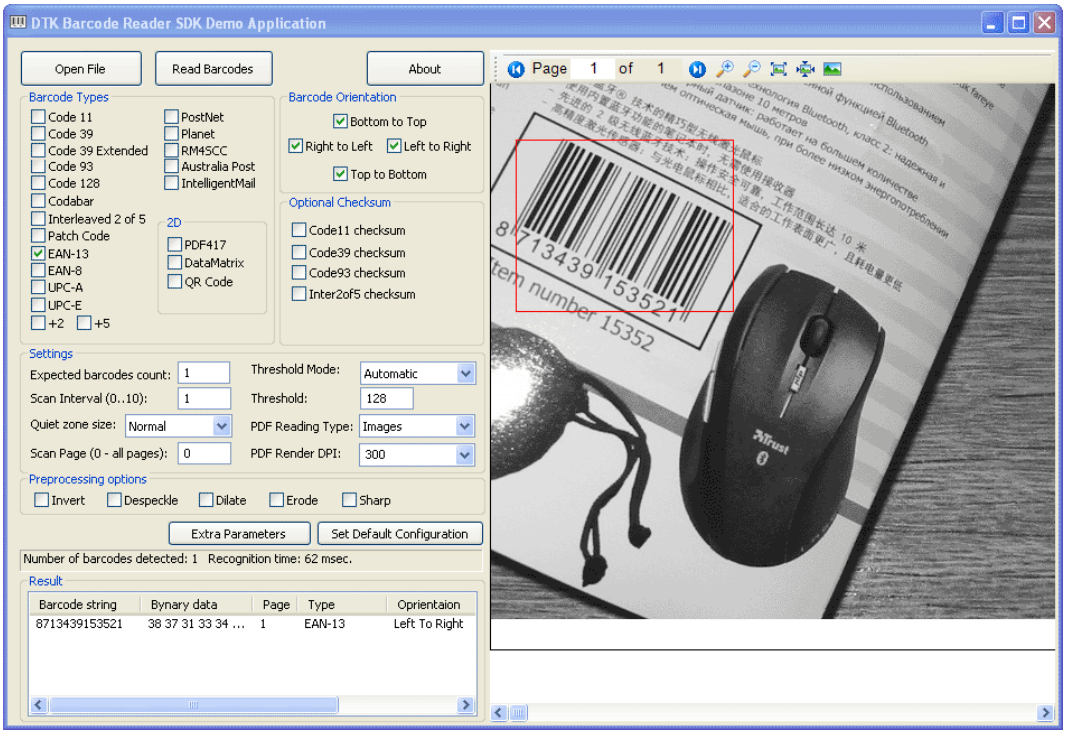
Vytvořená aplikace byla též porovnána s komerčními aplikacemi. Jednalo se o aplikace *DTK Barcode Reader*[2] (viz obr. 6.9) a *Softek Barcode Reader*[3] (viz obr. 6.10). Jelikož aplikace oplývaly velkým množstvím nastavení, tak bylo pro jednoduchost testování ponecháno nastavení s výchozími hodnotami. Drobná změna v nastavení se týkala pouze aplikace Softek Barcode Reader, kde výchozí hodnota tolerance natočení čárového kódu byla 5°. Aby tato aplikace nebyla znevýhodněna, hodnota tolerance natočení byla nastavena na maximální možnou, a to 45°.

Výsledky úspěšnosti a chybovosti na testovací sadě 500 snímků se nachází v tabulce 6.11. Pro objektivní posouzení byly do tabulky zahrnuty hodnoty, jež aplikace dosahovala při nejvyšší úspěšnosti a nejnižší chybovosti dle náležitých parametrů dekódování. Nechybí zde ani průměrné hodnoty ze všech provedených testů.

Tabulka 6.11: Srovnání výsledků s komerčními aplikacemi.

Aplikace	**Úspěšnost** [%]	**Chybovost** [%]
vlastní (úspěšnost)	74,6	9,4
vlastní (chybovost)	54,8	1,6
vlastní (průměr)	66,6	6,4
DTK Barcode Reader	60,4	2,3
Softek Barcode Reader	72,0	7,2

[2] Aplikaci lze nalézt na http://www.dtksoft.com
[3] Aplikaci lze nalézt na http://www.softeksoftware.co.uk



Obrázek 6.9: Aplikace DTK Barcode Reader.

Obrázek 6.10: Aplikace Softek Barcode Reader.

Kapitola 7

Závěr

Tato práce pojednává o dekódování čárového kódu v obraze. Představuje kategorie čárových kódů a seznamuje čtenáře s některými důležitými vlastnostmi těchto kódů. Dále popisuje dosavadní přístupy k dekódování čárového kódu.

Jelikož práce byla zaměřena na dekódování v reálném čase, byl kladen nejvyšší důraz na nízkou výpočetní náročnost. Z tohoto důvodu byl navržen efektivní lokalizační algoritmus založený na hledání podobných význačných bodů čárového kódu mezi skenovanými řádky. Vyšší úspěšnost dekódování zajišťovala binarizace dynamickým prahováním se subpixelovou přesností na skenovaných řádcích.

K testovacím účelům byla vytvořena kvalitní testovací sada čítající 500 snímků s čárovými kódy včetně anotací a také videosekvence pro otestování časové náročnosti. Aplikace byla řádně otestována jako celek a také i její jednotlivé části (lokalizace a dekódování) včetně provedené analýzy vzniklých chyb. V závěrečné části testování je věnován prostor porovnání výsledků s komerčními aplikacemi. Dle provedených testů lze tvrdit, že vytvořená aplikace je rovnocenným konkurentem komerčních aplikací. Má také uspokojivou časovou náročnost na zpracování jednoho snímku a je tedy možné uvažovat o dekódování čárových kódů v obraze v reálném čase.

Další možné rozšíření této práce by mohlo spočívat v možnosti dekódování více typů lineárních čárových kódů a také ještě více maximalizovat úspěšnost a minimalizovat chybovost aplikace. Poněvadž roste zájem o dekódování čárových kódů v terénu, nabízí se též možnost portovat aplikaci do mobilního telefonu.

Literatura

[1] Aas, K.; Eikvil, L.: Decoding Bar Codes from Human-Readable Characters. 1996.

[2] Adair, J.: Locating, Tracking, and Interpreting Ean-13 Bar Code Waveforms in a Two-Dimensional Video Stream. `http://www.acm.org/src/subpages/gf_entries_06/JeffreyAdair_src_gf06.pdf`.

[3] Adelmann, R.; Langheinrich, M.; Floerkemeier, C.: Toolkit for Bar Code Recognition and Resolving on Camera Phones - Jump Starting the Internet of Things. In *GI Jahrestagung (2)*, 2006, s. 366–373.

[4] Amould, S.; Awcock, G. J.; Thomas, R.: Remote bar-code localisation using mathematical morphology. In *Image Processing and Its Applications, 1999. Seventh International Conference on (Conf. Publ. No. 465)*, ročník 2, Manchester, UK, 1999, ISBN 0-85296-717-9, s. 642 – 646.

[5] Ando, S.; Hontani, H.: Automatic visual searching and reading of barcodes in 3-D scene. In *Vehicle Electronics Conference, 2001. IVEC 2001. Proceedings of the IEEE International*, Tottori, Japan, 2001, ISBN 0-7803-7229-8, s. 49 – 54.

[6] Bradski, G.; Kaehler, A.: *Learning OpenCV*. O'Reilly Media Inc., 2008.

[7] Chai, D.; Hock, F.: Locating and Decoding EAN-13 Barcodes from Images Captured by Digital Cameras. 2005, s. 1595–1599.

[8] Chen, Y.; Yang, Z.; Bai, Z.; aj.: Simultaneous Real-Time Segmentation of Diversified Barcode Symbols in Complex Background. *Intelligent Networks and Intelligent Systems, International Workshop on*, 2008: s. 527–530.

[9] Fang, X.; Wu, F.; Luo, B.; aj.: Automatic Recognition of Noisy Code-39 Barcode. In *ICAT '06: Proceedings of the 16th International Conference on Artificial Reality and Telexistence–Workshops*, Washington, DC, USA: IEEE Computer Society, 2006, ISBN 0-7695-2754-X, s. 79–82.

[10] Gallo, O.; Manduchi, R.: Reading Challenging Barcodes with Cameras. In *Winter Vision Meetings, Workshop on Applications of Computer Vision*, Snowbird, Utah, 2009.

[11] Jain, A. K.; Chen, Y.: Bar Code Localization Using Texture Analysis. In *Document Analysis and Recognition, 1993., Proceedings of the Second International Conference on*, Tsukuba Science City, Japan, 1993, ISBN 0-8186-4960-7, s. 41–44.

[12] Joseph, E.; Pavlidis, T.: Waveform recognition with application to bar codes. In *Proc. IEEE SMC 1991 Conf.*, Charlottesville, VA, USA: IEEE Computer Society, 1991, s. 129 – 134.

[13] Joseph, E.; Pavlidis, T.: Bar Code Waveform Recognition Using Peak Locations. *IEEE Trans. Pattern Anal. Mach. Intell.*, ročník 16, č. 6, 1994: s. 630–640, ISSN 0162-8828.

[14] Liao, H.-Y.; Liu, S.-J.; Chen, L.-H.; aj.: A Bar-code Recognition System Using Backpropagation Neural Networks. *Engineering Applications of Artificial Intelligence*, ročník 8, č. 1, 1995: s. 81 – 90.

[15] Lu, X.; Fan, G.; Wang, Y.: A Robust Barcode Reading Method Based on Image Analysis of a Hierarchical Feature Classification. In *IROS*, IEEE, 2006, s. 3358–3362.

[16] Ohbuchi, E.; Hanaizumi, H.; Hock, L. A.: Barcode Readers using the Camera Device in Mobile Phones. In *CW '04: Proceedings of the 2004 International Conference on Cyberworlds*, Washington, DC, USA: IEEE Computer Society, 2004, ISBN 0-7695-2140-1, s. 260–265.

[17] Otero, A.: A Robust Software Barcode Reader Using the Hough Transform. In *ICIIS '99: Proceedings of the 1999 International Conference on Information Intelligence and Systems*, Washington, DC, USA: IEEE Computer Society, 1999, ISBN 0-7695-0446-9, str. 313.

[18] Palmer, R. C.: *The Bar Code Book*. Helmers Publishing, 2001, ISBN 0-911261-13-3, 395 s.

[19] Žára, J.; Beneš, B.; Sochor, J.; aj.: *Moderní počítačová grafika*. Brno: Computer Press, 2004, ISBN 80-251-0454-0, 609 s.

[20] Shams, R.; Sadeghi, P.: Bar Code Recognition in Highly Distorted and Low Resolution Images. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, ročník 1, Honolulu, HI, Duben 2007, s. I–737–I–740.

[21] Shellhammer, S. J.; Goren, D. P.; Pavlidis, T.: Novel signal-processing techniques in barcode scanning. *Robotics & Automation Magazine*, ročník 6, 1999: s. 57 – 65, ISSN 1070-9932.

[22] Swartz, J.; Wang, Y. P.: Fundamentals of Bar Code Information Theory. *Computer*, ročník 23, č. 4, 1990: s. 74–86, ISSN 0018-9162.

[23] Tekin, E.; Coughlan, J.: A Bayesian Algorithm for Reading 1D Barcodes. In *CRV '09: Proceedings of the 2009 Canadian Conference on Computer and Robot Vision*, Washington, DC, USA: IEEE Computer Society, 2009, ISBN 978-0-7695-3651-4, s. 61–67.

[24] Tropf, A.; Chai, D.: Locating 1-D Bar Codes in Dct-Domain. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, Toulouse: IEEE, 2006, ISBN 1-4244-0469-X, ISSN 1520-6149, s. 741 – 744.

[25] Viard-Gaudin, C.; Normand, N.; Barba, D.: A bar code location algorithm using a two-dimensional approach. In *Document Analysis and Recognition, 1993., Proceedings of the Second International Conference on*, IEEE, 1993, s. 45 – 48.

[26] Wachenfeld, S.; Terlunen, S.; Jiang, X.: Robust recognition of 1-D barcodes using camera phones. In *ICPR08*, 2008, s. 1–4.

[27] Wang, K.; Zou, Y.; Wang, H.: 1D bar code reading on camera phones. *International Journal of Image and Graphics*, ročník 7, č. 3, 2007: s. 529–550, ISSN 0219-4678.

[28] Youssef, S. M.; Salem, R. M.: Automated barcode recognition for smart identification and inspection automation. *Expert Syst. Appl.*, ročník 33, č. 4, 2007: s. 968–977, ISSN 0957-4174.

Příloha A

Obsah CD

- Aplikace ve spustitelné formě
- Zdrojové kódy aplikace
- Programová dokumentace
- Manuál k aplikaci
- Plakát pro prezentaci práce
- Testovací sada čárových kódů
- Technická zpráva ve formátu PDF
- Zdrojové kódy technické zprávy (LaTeX)